More Than Digital Dirt:

Preserving Malware in Archives, Museums, and Libraries

by

Jonathan Farbowitz

A thesis submitted in partial fulfillment

of the requirements for the degree of

Master of Arts

Moving Image Archiving and Preservation Program

Department of Cinema Studies

New York University

May 2016

Table of Contents

Chapter 1: Why Collect Malware?[1]

Computer viruses are almost as old as personal computers themselves, and their evolution was only hastened by the birth of the internet. Within each code is a story about its author, about the time it was written, and about the state of computing when it wrought havoc upon our hard drives. —Attila Nagy, "14 Infamous Computer Virus Snippets That Trace a History of Havoc"

I want to talk about the problem of ensuring that this new medium will have a history, one that future scholars can write about critically. —Henry Lowood, "Shall We Play a Game"

This project discusses a multitude of concerns and challenges in the collection, preservation, and exhibition of malware and malware-infected digital artefacts for historical and cultural study. Malware is much more than a curiosity or an annoyance to be dispensed with and this thesis will reveal many of the narratives that malware presents in the history of computing and the internet.

Despite (or perhaps because of) its ubiquity online and on individual computers, malware has been largely ignored as a potential collection item within libraries, archives, and museums. Critically, this thesis seeks to develop a more nuanced discussion about how cultural heritage institutions should regard malware. These institutions must understand how malware infections affect their workflows for processing born-digital artefacts like hard drives and floppy disks. In addition, some cultural heritage institutions ought to view malware as a potential collection item. While Henry Lowood's statement above refers to the preservation of video games, along similar lines, I want to ensure that the medium of malware "will have a history."

---

[1] Portions of this thesis have been adapted from my previous paper, *Preserving Malware: A Case Study of the WANK Worm*.

This work serves as a preliminary blueprint for future research and analysis, and should not be taken as conclusive on the topics of malware collecting or best practices for malware preservation. My research focuses predominantly on computer worms and viruses; however, I will discuss other kinds of malware, including backdoors, ransomware, and spyware where relevant.

A cultural heritage institution intentionally collecting and preserving malware could provide immense benefits to scholars in the humanities and sciences as well as the public at large. However, in my research, I have not yet encountered a cultural heritage institution in the United States that is committed to systematically collecting and preserving a historical sample of viruses, worms, or any other kind of malware. Computer and technology museums range from not collecting any malware at all, to just beginning to address the idea.[2]

Through their exhibitions and collections, technology museums and other institutions that host exhibits on the history of computing tend to present a narrative of linear progress in the development of software and other digital technologies. These narratives often lionize individuals, software companies, or hardware manufacturers. However, these institutions could present the full spectrum of how technology is used and exploited by people throughout the world, particularly in ways unintended by hardware or software companies—they could make

---

[2] Chris Avram, "Re: Research on Malware," March 2, 2015. While the Computer History Museum in Mountain View, California has several editions of antivirus software in its collection, the only born-digital malware artefact listed in its public catalog is an original disk that contains the source code for the Morris Worm (1988). However, the museum has books as well as video documentation of lectures related to malware. Among the few cultural heritage institutions that hold items related to malware, The Museum of Modern Art has a copy of the Newton "Virus," created by the Troika art collective, in its architecture and design collection. However, Newton is somewhat of a "hoax virus." It takes a screenshot of a user's desktop in order to mimic the desktop icons falling, but has no other effects and does not replicate like a virus.

room for exploring uses of computing considered deviant or antisocial, or where anonymous

individuals or groups are responsible, such as malware programming.

Computer security and antivirus companies as well as governmental organizations save

malware, but these collections have not been widely accessible and likely do not have the kind of

curation and attention to the needs of humanities and social science researchers that cultural

heritage institutions could provide. Accessing these collections can also be challenging.

Computer security companies and governments also have no mandate for the long-term

preservation of these collections or to collect related contextual or ancillary materials (for

example, articles about malware, websites that discuss it, screen captures, and video

walkthroughs of infections) that would allow researchers to explore the cultural, sociological,

and political intricacies of malware development and infection.

Defining Malware

This section will place the term "malware" in its proper context and define the key

characteristics of several kinds of malware: computer viruses, worms, and spyware. The word

"malware" is a portmanteau of the words "malicious" and "software" and is generically defined

as "any software used to disrupt computer operation, gather sensitive information, or gain access

to private computer systems."[3] Two characteristics common to most malware are that it operates

without the consent of the computer user or network administrator, and that much of it

self-replicates in order to spread. Computer security researcher Vesselin Bontchev has said that a

---

[3] "Malware," *Wikipedia*, accessed May 13, 2015, http://en.wikipedia.org/wiki/Malware.

virus ruins "the trust that the user has in his/her machine because it causes the user to lose his or her or belief that she or he can control this machine."[4]

Nevertheless, self-replication, autonomous behavior, or the absence of user consent may be necessary, but not sufficient to indicate malicious code. Thus, "if the emphasis is placed on reproduction routines, virus- and worm-like programs cannot be said to be malicious by definition."[5] New media theorist Jussi Parikka argues that, "essentially the same program can be defined as a utility program in one context and as a malware program in another...many basic utility programs have for years been virus-like, even though such programs often require the consent of the user to operate."[6] Many maintenance routines or automatic software updates in operating systems like Microsoft Windows or Mac OS X run independently, often without the user's knowledge or explicit consent, yet no one would question their legitimacy or intent and the mainstream computer security community would never classify them as malware.

The label "malware" applied to certain software or scripts is by no means stable or unproblematic. Several writers, including Parikka, have pointed to the instability of the term. Software defined as "malware" must always be understood in the context of who classifies it—one person's "malware" could be another person's tool for research; deliberate warning to software developers, computer security companies, or users; method of civil disobedience; law enforcement or surveillance technology; cyberweapon; or software-based artwork.

Despite the instability of the term, the antivirus and computer security industry control the most widespread definitions of malware—and these definitions are colored by an industry

---

[4] Quoted in Jussi Parikka, *Digital Contagions: A Media Archaeology of Computer Viruses*, Digital Formations, v. 44 (New York: Peter Lang, 2007), 35.
[5] Parikka, *Digital Contagions*, 19.
[6] Ibid., 51.

primarily concerned with the interests of its biggest clients: businesses and governments. Thus, anything perceived as threatening to the smooth functioning of commerce or governance will get designated as malware. Furthermore, corporate and government interests have already shaped the very discourse that surrounds this kind of software: "The visibility of viruses as harmful software programs has been generated...through the interests of national security as well as international business—to which computer viruses represent a disruption in the global flows of capital."[7] Ultimately, as Parikka argues, the classification of virus- and worm-like programs as malicious threats has been historically contingent: "the incorporeal morphing of these programs into malicious creatures was linked to the increasing importance software and network computing played in a post-Fordist culture."[8] The definition of malware has always been shaped in relationship to the increasing dependence on information technologies as an engine of capitalist expansion.

Despite my desire to destabilize conventional wisdom about what constitutes "malicious software," from a preservation standpoint, what can be established is that the label "malware" refers to a wide variety of different pieces of software and scripts that have diverse mechanisms of operation when executed, affect different files or operating systems, and present differing degrees of risk to individual computers or networks. In art conservation terms, two different pieces of malware may have entirely different "significant properties."[9] As such, decisions about

---

[7] Jussi Parikka and Tony D. Sampson, eds., *The Spam Book: On Viruses, Porn, and Other Anomalies from the Dark Side of Digital Culture*, Hampton Press Communication Series : Communication Alternatives (Cresskill, N.J: Hampton Press, 2009), 120.
[8] Parikka, *Digital Contagions*, 51.
[9] Significant properties are "those properties of digital objects that affect their quality, usability, rendering, and behaviour." See Hedstrom, Margaret L., Christopher A. Lee, Judith S. Olson, and Clifford A. Lampe. "'The Old Version Flickers More': Digital Preservation from the User's Perspective." *The American Archivist* 69 (Spring/Summer 2006): 159–87.
Significant properties can include characteristics like the resolution, color palette, or timing of a work of art.

preserving individual pieces of malware or assessing infections may need to be made on a case-by-case basis, as is often done with the complex and interrelated components of a time-based media artwork.

A computer virus is software that, when executed, inserts copies of itself into other programs, other files, or the boot sector of a hard drive or disk.[10] A virus can be an executable program, a script, or even a Microsoft Word document that when opened runs a macro.[11] Viruses can even look like innocuous files (such as a JPEGs or GIFs) but when opened execute viral code. A virus often appends its code onto the code of other programs like Adobe Photoshop so that when the user runs Photoshop, the virus's code runs as well and the virus may continue to copy itself to other locations on the user's hard drive or disk.

Viruses often have adverse effects on infected computers, which could range from displaying graphics or a message to the user, to the deletion of files or the erasure of a computer's BIOS[12] so that it can no longer boot. Often the word "payload" is used to describe the effects of a virus or other piece of malware, and the term "payload screen" is used to describe graphics or messages displayed on infected computers. While some viruses have adverse effects, one must take into account that "not all computer viruses are destructive...certainly computer viruses can delete data, but they can also be performative (e.g., demonstrating a security violation), exploratory (e.g., gaining access), or based on disturbance rather than destruction

---

[10] "Computer Virus," *Wikipedia*, accessed May 13, 2015, http://en.wikipedia.org/wiki/Computer_virus.
[11] A macro is a stored command or series of keystrokes, which can be activated later by pressing a single key. The use of macros can make completing certain tasks in Microsoft Word more efficient.
[12] "BIOS (basic input/output system) is the program a personal computer's microprocessor uses to get the computer system started after you turn it on." See http://whatis.techtarget.com/definition/BIOS-basic-input-output-system.

(e.g., rerouting network traffic, clogging network bandwidth)."[13] Using a malware to demonstrate a security violation or vulnerability is often known as a "proof of concept."[14]

A computer worm is a like a virus except that the worm is autonomous and can self-replicate without attaching itself to another program or file.[15] Once released onto a network, worms can spread without further human intervention from computer to computer. While some worms do not have a destructive payload, their self-replicating nature can slow networks down to a crawl. Worms can replicate especially quickly over the internet. For example, the SQL Slammer Worm (2003) reportedly infected around 75,000 computers, most of them in the first ten minutes of its release.[16]

A worm often installs a backdoor[17] in an infected computer so that another person, often the worm's creator, can control the infected computer remotely through a command-and-control server without the knowledge of the infected computer's user. When many infected computers are under remote control this is referred to as a botnet. Hackers will often rent out their botnets for a fee so that others can use them as a platform to send spam or attack websites and network infrastructure.

Spyware is software that monitors and logs incoming or outgoing network traffic, keystrokes, or other information from a computer without the user's knowledge or consent. The

---

[13] Alexander R. Galloway and Eugene Thacker, *The Exploit: A Theory of Networks*, Electronic Mediations, v. 21 (Minneapolis: University of Minnesota Press, 2007), 83.
[14] Individuals who release proofs of concept sometimes see themselves as doing a public service by calling attention to vulnerabilities in software or operating systems and potentially shaming a software developer into fixing them. This is especially true if the proof of concept does not damage computers or compromise individual's personal information in the process. Proofs of concept may also be published as white papers.
[15] Suelette Dreyfus and Julian Assange, *Underground: Tales of Hacking, Madness, and Obsession on the Electronic Frontier* (Kew, Australia: Mandarin, 1997), 20.
[16] "SQL Slammer," *Wikipedia*, accessed May 1, 2015, https://en.wikipedia.org/wiki/SQL_Slammer.
[17] Backdoors are security vulnerabilities that bypass normal authentication (such as password protection), or allow unauthorized access to a computer systems.

collected information is sent to other systems for later analysis, or the target computer can be

monitored in real time. Spyware has been used by commercial entities to monitor their

customers' habits, but has also been used by governments to surveil citizens.

The use of spyware has increased significantly since the beginning of the twenty-first

century. In 2001, journalists revealed that the FBI had developed keystroke logging software

called Magic Lantern that could be installed via a deceptive email attachment.[18] Gamma

International, based in Germany and Gamma Group, based in the United Kingdom, develop and

sell spyware called FinFisher (for desktop and laptop computers) and FinSpy (for mobile

devices) to law enforcement and intelligence agencies. FinFisher can be installed on a target

computer through infected email attachments or fake software updates and allows authorities to

monitor all processes on a target computer and even use the computer's front-facing camera to

surveil the user.
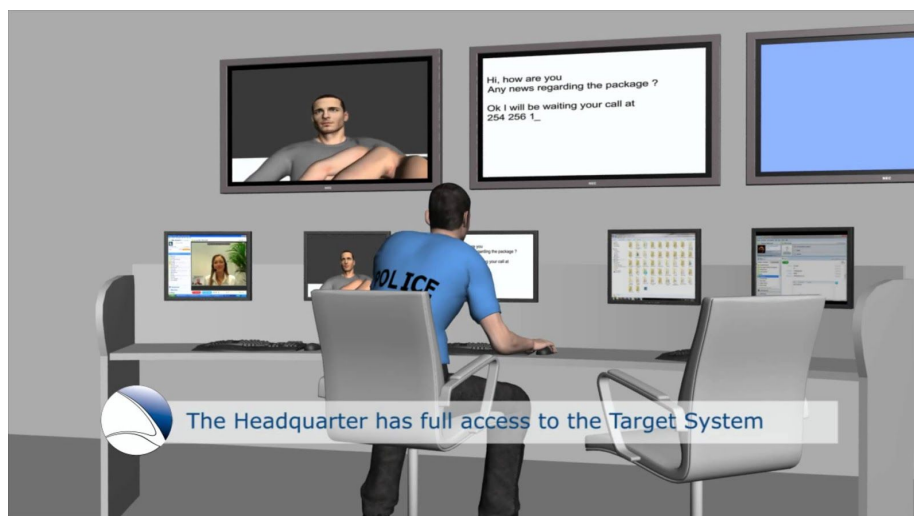
Figure 1.1: Still from a FinFisher promotional video. A police officer uses FinFisher spyware to monitor a "target system." The screen on the top left represents the view of the front-facing camera above a computer's screen. The white screen in the middle represents the target's chat messages being monitored. **(**WikiLeaks -

---

[18] Sullivan, Bob. "FBI Software Cracks Encryption Wall." *NBC News*, November 20, 2001. http://www.nbcnews.com/id/3341694/ns/technology_and_science-security/t/fbi-software-cracks-encryption-wall/#. VyWF9KODGko.

https://wikileaks.org/spyfiles4/documents/FinSpy-Video.wmv)

Evidence of active FinFisher servers has been found in twenty-five countries including the

United States.[19] While Gamma International markets their software as a tool for law

enforcement, the company has received criticism from human rights groups who argue that

authorities have used FinFisher to target political opponents, activists, and human rights

observers.[20] Through collecting spyware or other kinds of malware employed by governments,

cultural heritage institutions can preserve evidence of state repression, including information

about what tools were at the government's disposal and how these tools were used. The existence

of organizations such as The Citizen Lab and Equality Labs and the reporting of organizations

such as the Electronic Frontier Foundation and the American Civil Liberties Union indicate that

there is significant interest in researching and analyzing spyware outside of the computer

security community.

Why Preserve Malware?

> *Computer viruses deserve a museum: they're an art form of their own* – Title of a 2016
> article by Jussi Parikka

Malware is a form of cultural heritage and an important part of the historical record. This

section examines the many unique uses of malware as an art object, tool for activism, and

cyberweapon. Computer security expert Mikko Hypponen states that "many old-school virus

---

[19] The other countries where Finfisher servers has been found include: Australia, Bahrain, Bangladesh, Brunei, Canada, Czech Republic, Estonia, Ethiopia, Germany, India, Indonesia, Japan, Latvia, Malaysia, Mexico, Mongolia, the Netherlands, Qatar, Serbia, Singapore, Turkmenistan, United Arab Emirates, the United Kingdom, and Vietnam.
[20] Marquis-Boire, Morgan, Bill Marczak, Claudio Guarnieri, and John Scott-Railton. "You Only Click Twice: FinFisher's Global Proliferation." *The Citizen Lab*, March 13, 2013. https://citizenlab.org/2013/03/you-only-click-twice-finfishers-global-proliferation-2/.

writers were using their viruses as a means of expression. That's why we get all these displays of

animations, sound, and pictures. Some would call it art."[21] He further argues that historical

viruses represent an "important chapter of [the] internet's history."[22] Jussi Parikka writes that

> Creating viruses became an important subculture and part of new sorts of cultural activities, practices and interests. We too often think that all malware is by necessity just vandalism or criminal activity. The actual skills of coding them...may be just a hobby for some but an art form for others. And viruses themselves are cultural objects that tell the story of contemporary security.[23]

> Furthermore, malware programming contributes to the overall history of software as

viruses "often involve innovative programming techniques that have been used in other areas of

computer science."[24] As just one example, digital media scholar Finn Brunton points to a 1982

paper by John Shoch and Jon Hupp about "Worm Programs":

> Shoch and Hupp were envisioning something quite inventive, particularly for the time: a 'distributed computation'...a single program operating across many machines and taking advantage of idle processing power to do its work. This 'worm' is the first monster from which the others spring with the same essential DNA, the worm that grows at night...as it segments individual underused machines for a collective purpose.[25]

Jane Gruning, doctoral candidate at University of Texas at Austin's School of Information,

argues that "viruses are not only an important part of hacker culture, but they have also affected

the ways in which we use the Internet and share digital objects."[26] The existence of malware has

profoundly affected the way software developers design programs and operating systems. The

spread of malware has caused generations of computer users to skeptically examine the contents

---

[21] Quoted in Claire Voon, "A Museum for the Blocky Graphics of Early Computer Viruses," *Hyperallergic*, February 18, 2016, http://hyperallergic.com/274139/a-museum-for-the-blocky-graphics-of-early-computer-viruses/
[22] Ibid.
[23] Jussi Parikka, "Computer Viruses Deserve a Museum: They're an Art Form of Their Own," *The Conversation*, February 19, 2016,
https://theconversation.com/computer-viruses-deserve-a-museum-theyre-an-art-form-of-their-own-54762.
[24] Galloway and Thacker, *The Exploit*, 83.
[25] Brunton, Finn. *Spam: A Shadow History of the Internet*. Cambridge, Mass.: MIT Press, 2013.
[26] Jane Gruning, "Rethinking Viruses in the Archives," (Poster, Archival Education and Research Institute, 2012), https://www.ischool.utexas.edu/~janegru/images/Gruning_AERI2012.pdf.

of unidentified floppy disks or thumb drives, or to avoid opening unknown email attachments

(some email services will not even allow the user to open attachments without scanning them

first). The ways in which malware has affected web browsers may be invisible to users (such as

sandboxing).[27] Others are more visible, such as warnings that prevent users from visiting

potential attack sites. The creation of windows that ask "This item is downloaded from the

internet. Are you sure you want to open it?" and the move toward company-controlled app stores

for downloading software were both influenced by lessons learned from past malware epidemics.

Video preservationist Jim Lindner contends that the content of the world's videotapes

exemplifies the "texture of who we are, the tapestry of who we are,"[28] because videotape

captured both quotidian moments and those considered historically important. Malware is part of

the texture of digital life because while it does occasionally create headlines, it is also "a

pervasive feature of the internet"[29]—something malware coders create daily, computer users

encounter routinely. The threat of malware has spawned a multi-billion-dollar security industry

concerned with defending against new attacks that are carried out every minute. As of the early

2000s, malware coding has become a service for hire procured by governments and corporations.

In the last several years, the use of malware for political purposes has continued to increase.

If malware were not preserved, a significant portion of contemporary computer users'

experiences as well as the "texture" of the internet and of computing itself would be lost.

Malware can express particular cultural anxieties (as seen in the many computer viruses that

---

[27] Sandboxing is a method for executing programs or scripts within an isolated environment (a sandbox) that does not have full access to a computer's resources. Modern web browsers employ some degree of sandboxing to prevent malware from controlling a computer.

[28] TIME. "Game Changers: Jim Lindner, Archive Automator." *YouTube*, March 23, 2012. https://www.youtube.com/watch?v=b8QvfimOfko.

[29] Maureen Pennock, "Web Archiving" (Digital Preservation Coalition, March 2013), http://dx.doi.org/10.7207/twr13-01.

reference HIV/AIDS) and political visions (for example, the WANK Worm, released in protest

of the launch of the Galileo Space Probe, and COFFSHOP.COM, which advocates for the

legalization of marijuana). It can also be used as a tool to punish resistance to the state (for

example, pro-government Twitter bots used in Mexico that jammed activists' hashtags and

tweeted death threats).[30] The multifaceted motivations of malware programmers will be

discussed in more detail in Chapter 2.

The texture of life must also include the day's frictions and disruptions and "a virus is a

disruption to the everyday logic and rhythm of the social order, a catastrophe...a virus can

translate a portion of technical code into repercussions across scales from economics to politics."

[31] ILOVEYOU infected tens of millions of computers and cost billions of dollars to remove,[32]

and the 2007 hacking incident in Estonia took government and banking websites offline.[33]

Malware's remarkable ability to rapidly disrupt the "rhythm of the social order" and the

wide-ranging consequences of its release ought to attract the interest of any social or political

historian of the twentieth and twenty-first centuries or those researching the history of

catastrophes.

Within software preservation, malware may currently occupy a similar position to that of

orphan or ephemeral audiovisual works like industrial films, commercials, and home

movies—works that individuals, like prominent archivist Rick Prelinger, were vigorously

questioned for collecting decades ago. However, the current consensus among archivists and

---

[30] Finley, Klint. "Pro-Government Twitter Bots Try to Hush Mexican Activists." *WIRED*, August 23, 2015. https://www.wired.com/2015/08/pro-government-twitter-bots-try-hush-mexican-activists/.

[31] Parikka, *Digital Contagions*, 7.

[32] Garza, George. "Top 10 Worst Computer Viruses." *Catalogs.com*. Accessed July 23, 2017. http://www.catalogs.com/info/travel-vacations/top-10-worst-computer-viruses.html.

[33] Davis, Joshua. "Hackers Take Down the Most Wired Country in Europe." *WIRED*, August 21, 2007. https://www.wired.com/2007/08/ff-estonia/.

historians is that these audiovisual works are indeed valuable for research and worth saving.

Many archives now consider acquiring industrial films or home movies part of their collecting

policy.

Some institutions have already considered the potential for future research on malware.

For example, in her report on web archiving, Maureen Pennock notes that "many archives

choose to scan harvests and identify malware but prefer not to exclude or delete them from ingest

into their repositories, as exclusion threatens the integrity of a site and their prevalence across the

web is a valid research interest for future users."[34]

A growing body of literature has emerged—at the intersection of media and

communication studies, computer science, science and technology studies, history, and

sociology—that considers the social implications of malware. Such work includes books by

Alexander Galloway, Eugene Thacker, Finn Brunton, Tony Sampson, and Jussi Parikka. This

group of scholars are interested in what "anomalous digital objects" (including malware and

spam) reveal about the history of computing and contemporary culture. Parikka argues that

"Viruses, etc. are one way of understanding what happens in network culture."[35] In *The Spam

Book*, Parikka and Sampson argue that "however intrusive and objectionable...the digital

anomaly has become central to contemporary communication theory."[36]

Studying malware can help researchers understand the capabilities, vulnerabilities, and

failings of operating systems or pieces of commercial or mainstream software.[37] Galloway and

Thacker assert that tracing malware epidemics actually helps one understand how computer

---

[34] Pennock, "Web Archiving," 14.
[35] Jussi Parikka, "RE: Query," February 19, 2016.
[36] Parikka and Sampson, *The Spam Book*, 3.
[37] This point was suggested in conversation with Martin Oberist.

networks function in a profound way. They argue that the "dissonance" between the "self-organizing qualities of emergent network phenomena" and "one's ability to superimpose a top-down control on that emergent structure" becomes "most evident in network accidents or networks that appear to spiral out of control—internet worms and disease epidemics, for instance. But calling such instances 'accidents' or networks 'out of control' is a misnomer. They are not networks that are somehow broken but *networks that work too well.*"[38]

Given this body of scholarship, cultural heritage institutions such as museums, archives, and libraries can consider collecting malware and malware-related materials to support this research with relevant, curated archival collections—research that will only become more difficult over time as historical malware disappears from the internet or from society's collective memory. Not only can institutions support existing scholarship, but through careful curation and collecting they can also "aid in articulating a scholarly research agenda."[39]

Histories Within Malware

Malware coding is entwined with histories of hacking, software development, political activism, art production, cybercrime, and state repression. If "hackers actively constitute themselves as a subculture through the performance of technology,"[40] then contained within malware code or the effects of a malware infection is a record of that performance. Douglas Thomas argues that representations of hackers in the media reveal anxieties about technology more than they do about the "culture of hackers or activity of hacking."[41] Perhaps a malware

---

[38] Galloway and Thacker, *The Exploit*, 5–6.
[39] Matthew G. Kirschenbaum et al., *Digital Forensics and Born-Digital Content in Cultural Heritage Collections*, CLIR Publication, no. 149 (Washington, D.C: Council on Library and Information Resources, 2010), 63.
[40] Douglas Thomas, *Hacker Culture* (Minneapolis: University of Minnesota Press, 2002), xx.
[41] Ibid.

archive would become an opportunity for scholars or the public to better understand the mindset

of individual computer hackers (through "primary source documents" like source code) and

derive broader observations about the culture of hacking. In addition, through public exhibition,

an institution can present the culture of hackers in a more nuanced and revealing way. A

collection that included a thoughtful balance of code and ancillary documentation could

reposition hacking "as a cultural, rather than technical, activity."[42]

Novel coding techniques or software concepts that originated in malware occassionaly

make their way into mainstream software. Brunton argues that "as a vein of quartz suggests the

possibility of gold nearby, so does spam often imply new areas of exploitation and innovation

online."[43] The same could be said for certain kinds of malware. In fact, the first anti-malware

procedures to eliminate computer worms from networks behaved exactly like worms themselves.

"Parasitic computing" projects, such as SETI@home, that link idle computers together to

perform computations on massive amounts of data are conceptually similar to a botnet, though

these projects require the consent of users.

Artists have coded and released malware at least since the 1980s, and their works have

many forerunners outside of the digital realm. Scholar Alan Liu describes Gustav Metzger's

artworks as displaying an

> early form of...'viral aesthetics.' This refers to an aesthetic in which the distinction
> between production and destruction is often blurred, revealing a 'a destructivity that
> attacks knowledge work through technologies and techniques internal to such work.' If
> Metzger is the industrial forerunner of viral aesthetics...the contemporary work of artists
> like Jodi and Critical Art Ensemble are its heirs.[44]

---

[42] Ibid., xxi.
[43] Brunton, *Spam*, 184.
[44] Galloway and Thacker, *The Exploit*, 108.

Malware not created as an intentional art object can generate aesthetic experiences as well. This fact has been acknowledged in the exhibition "I Love You," curated by digitalcraft and first exhibited at the Museum of Applied Arts in Frankfurt, Germany. The exhibit showcased malware intentionally created as art, such as Biennale.py (2001), alongside malware not expressly created for that reason, such as ILOVEYOU (2000). Malware coders can be viewed in relationship to artists who have experimented with noise and chance, such as John Cage, and as a source of inspiration for artists who work with digital glitches and mods such as Cory Arcangel. Malware as fine art will be discussed further in Chapter 2.

As the governments of nation-states are increasingly using malware, particularly spyware, to surveil and discipline their own populations, this kind of malware can also be preserved as evidence of state repression. Malware is increasingly becoming a weapon in what politicians and other commentators call "cyberwar." As Parikka aptly states, "the malware museums of the future will have to include the extensive measures taken by state intelligence agencies in the name of cyberdefense, with civilian casualties included. The problem is much of that data is likely to be secret, stored in the data centres and server farms of government agencies."[45]

As more countries rely on computer networks for critical infrastructure, cyberwar-like events involving malware have become more frequent:

> In April 2007, the Estonian government provoked an international incident by removing a bronze statue of a Soviet soldier from the center of Tallinn...Almost immediately thereafter, Estonia's network traffic started to surge. The servers for several major Estonian institutions, including government ministries, banks, and newspapers, were hit with massive spikes in activity, enough to eat up their bandwidth and repeatedly take them offline.[46]

---

[45] Parikka, "Computer Viruses Deserve a Museum."
[46] Brunton, *Spam*, 189.

The incident in Estonia involved several Distributed Denial of Service attacks (discussed in more detail below) enabled by malware. The malware allowed the attackers to organize computers into a botnet. The events were compared by politicians and the media (however hyperbolically) to "a digital Pearl Harbor" or a nuclear attack.[47] Such comparisons only underscore the extent to which computer networks have permeated the daily lives of many people worldwide, and have become essential for governments to function, as well as how seriously attacks on them are now taken.

Since at least the 1975 novel *The Shockwave Rider*, the release of malware has been envisioned as a form of political resistance. Leftist author John Brunner "imagines rebels releasing a 'tapeworm' to bring down the computer network of an authoritarian regime."[48] Douglas Thomas (along with Galloway and Thacker) believes that "the medium of the computer affords a particular avenue of resistance that speaks to broader questions of technology and culture."[49]

Since its very beginning, malware's payload screens have spread their own political messages. The MacMag Virus (1988) displays a message of "Universal Peace" as its payload and the MS-DOS virus COFFSHOP.COM (1990s) displays an image of a marijuana leaf and the statement "Legalize Cannabis." The Dukakis Virus (1988) promoted the candidacy of Michael Dukakis by displaying "Dukakis for President" on an infected computer. These viruses did not have an additional payload.

---

[47] Ibid.

[48] Jeff Sparrow and Jill Sparrow, *The Enemy within*, Radical Melbourne, Jeff Sparrow & Jill Sparrow ; 2 (Carlton North, Vic: Vulgar Press, 2004), 183.

[49] Thomas, *Hacker Culture*, xvii.
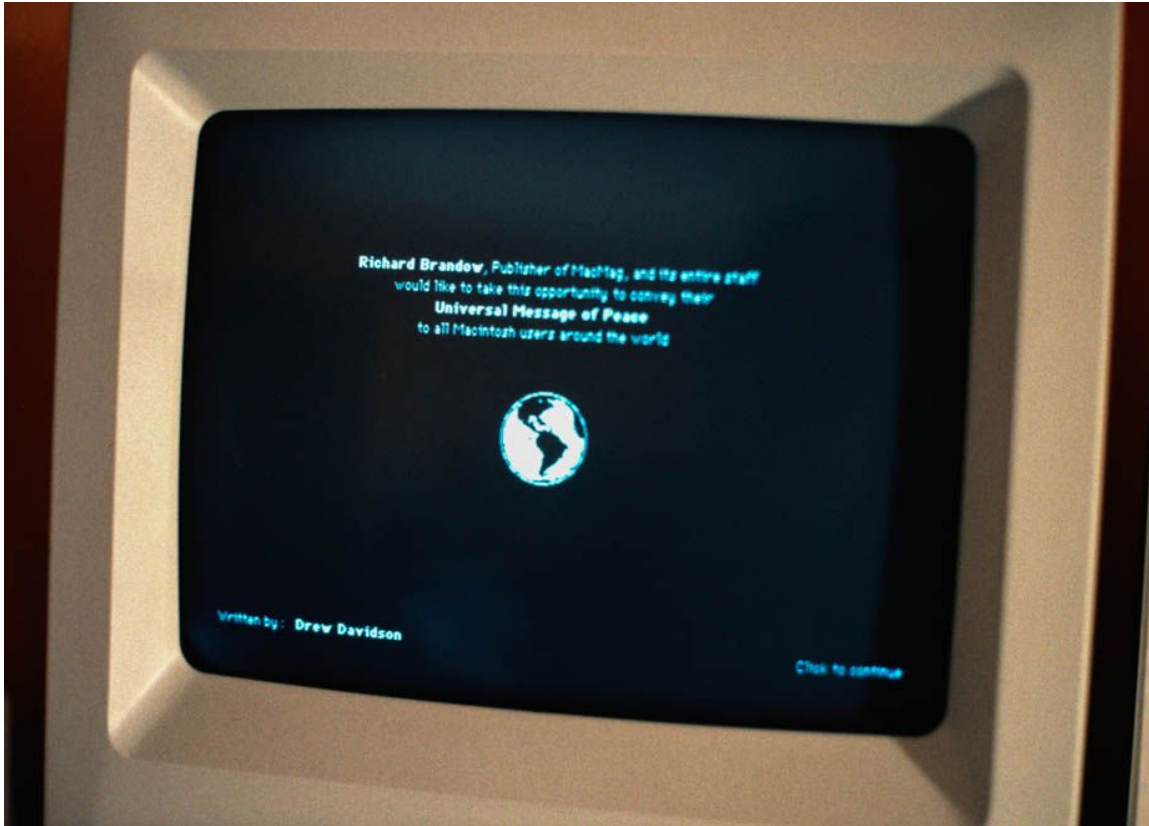
Figure 1.2: The payload screen of the MacMag Virus. (Wikipedia)

At the same moment, malware coders had more ambitious aims than infecting a few

personal computers. Worms Against Nuclear Killers (also known as the WANK Worm), released

in 1989, may be the first instance of "hacktivism," a portmanteau of "hacking" and "activism"

which signifies the manipulation of computer systems in the service of a particular political

cause.[50] The worm infected NASA's computers, presumably in protest of the imminent launch of

the Galileo Space Probe which contained a nuclear reactor and was opposed by anti-nuclear

activists who held protests at Kennedy Space Center.

---

[50] For more information on WANK and its potential preservation see Farbowitz, *Preserving Malware: A Case Study of the WANK Worm*.

Since WANK's release, offline political activism and hacking have continued to

converge with the development of specialized tools for online civil disobedience, often classified

as malware. As Brunton notes,

> The DDoS [Distributed Denial of Service, an attempt to overload a target website with traffic and take down its servers] has also made a strange lateral move into protest events, becoming the weapon of choice for online activist groups such as Anonymous. Programs including the grandly named 'Low Orbit Ion Cannon'...enable individuals who download it to voluntarily join a botnet. This public-spirited botnet can then be directed to attack sites like those of organizations that were hostile to WikiLeaks and of repressive governments like Syria's. The values of these technologies, and the narratives in which they can be enlisted, are in constant transformation.[51]

A malware collection that includes software used for hacktivism could assist researchers in

understanding and tracking the use of these technologies, and their connection to offline activism

over time.

The connection between hacking and political resistance is especially relevant to

Galloway and Thacker who directly connect developing network exploits with effective political

resistance, arguing that "*To be effective, future political movements must discover a new exploit.*"

[emphasis in original][52]

Entangled with the issue of malware preservation is also the issue of how computer code

is analyzed within the humanities. Certain critics consider code akin to a literature, which

presupposes a hypothetical "hermeneutics of code" that has yet to be developed.[53] This debate

highlights the importance of understanding the human aspect of programming including the

intentions, preferences, and personal quirks of the programmer as well as how the code relates to

other previously written programs. One can relate information about computer viruses to a

---

[51] Brunton, *Spam*, 192.
[52] Galloway and Thacker, *The Exploit*, 22.
[53] Any true "hermeneutics of code" would have to eschew looking at any use of language within the code (such as comments, etc.) and merely examine the structure and arrangement of the code itself for interpretive clues.

"wider environment of coding, not just the code itself but the ecologies, cultures of coding as a critical skill."[54]

Preserving a Working Environment

While preserving malware will help researchers explore larger societal trends, new perspectives arise by narrowing the focus to an individual's infected computer. Multiple publications discussing digital forensics refer to the importance of researchers analyzing "data associated with the authoring process and characteristics of the creator's working environment." [55] Given that "computers today function as personal environments and extensions of self" and the fact that "their desktops are a reflecting pool of our digital lives," archivists would benefit from considering a computer system and its files as "a physical environment replete with potential evidence."[56] In conceptualizing computers as brimming with potential clues, malware infections must be considered in turn for their evidentiary value. One question worth asking is whether a malware infection constitutes a salient characteristic of an individual's personal life or working environment.

The evidentiary nature of the infection becomes especially clear if the individual is an activist being spied on. Just as historians today examine Martin Luther King Jr.'s or Emma Goldman's FBI files, researchers of the future would certainly have some interest in knowing that an activist's computer was infected with spyware or a cyberweapon, which government

---

[54] Parikka, "RE: Query."

[55] Kam Woods, Christopher Lee, and Simson Garfinkel, "Extending Digital Repository Architectures to Support Disk Image Preservation and Access," 2011, http://www.ils.unc.edu/callee/p57-woods.pdf, 57. The point was also suggested in conversation with Finn Brunton.

[56] Kirschenbaum et al., *Digital Forensics and Born-Digital Content in Cultural Heritage Collections*, 7.

created it, and how the infection occurred. Additionally, they may want to know how the software worked and even see it demonstrated live on a computer.

Would a researcher be interested in knowing that a writer, artist, or intellectual had a malware infection on their computer that was not spyware? How would such a situation affect their work habits? Perhaps a writer started a novel or essay in Microsoft Word and their document was infected with a macro virus and could no longer be accessed safely. The infection may have caused them to either abandon a work, start the same project from scratch, or purchase a new computer—facts which would interest a researcher. In addition, a virus or worm's payload screen could have inspired a writer or artist in the same way that, for example, a newspaper article or a photograph may have.

A researcher studying an artist may want as much information as possible about the conditions under which they worked. If the artist used a computer as part of their practice, the operating system and programs constitute part of the artist's working environment. If an artist intentionally infected a computer for a work, or the artist was collecting malware for their own research purposes, or using malware as part of their work (such as James Hoff), saving that malware would be critical to understanding their work or their process.

Whether the malware infection becomes salient in the biography of a writer or artist may ultimately depend on the kind of infection or how the individual reacts. In an interview on *Conan*, George R. R. Martin states that he writes his novels on a computer with no internet connection running MS-DOS and Wordstar 4.0. His interview suggests that he uses these antiquated tools, in part, to avoid getting computer viruses.[57] Who knows what writing Martin

---

[57] "Conan." TBS, May 13, 2014. http://teamcoco.com/video/george-r-r-martin-dos-program.

may have lost to viruses in the past, which is perhaps why he chose this peculiar working

environment. Perhaps malware infections have claimed manuscripts from other prominent

writers and caused them to alter their working habits.

Other infections may have less direct relevance, such as if Salman Rushdie's computer

was used as part of a botnet without his knowledge. However, archivists may want to err on the

side of caution and assume that malware infections may be relevant to researchers at some point

in the future, which would require preserving the infected version of the born-digital artefact as

part of the original working environment.

Current Malware Collection Efforts

Many organizations and individuals already collect malware: computer security

companies, individual security researchers, and an active amateur collector community on the

internet that hosts malware repositories (including "TheZoo," a malware collection on GitHub[58])

and posts on websites such as VX Heaven. Unfortunately amateur malware repositories can be

highly disorganized, redundant, and filled with non-malware items.[59]

The United States Computer Emergency Response Team Coordination Center

(CERT/CC), a federally-funded research center that works to improve the security of software

and the internet, has been formally collecting malware since 2001. Ed Stoner, technical director

of threat analysis at CERT/CC, states that "Nearly all of our collection is a result of other groups

collecting artifacts and giving it to us. Historically, most of those groups were collecting

malware through incident response operations. That's changed over time as various researchers

---

[58] See https://github.com/ytisf/theZoo
[59] See for example Vesselin Bontchev, "Analysis and Maintenance of a Clean Virus Library," *VX Heaven*, 1993, http://vxheaven.org/lib/avb01.html.

and for-profit companies have increased interest in the space."[60] CERT/CC has previously made samples available to researchers and government agencies. Most researchers accessing the collection have engineering or computer science backgrounds, but there has been a recent uptick in researchers requesting materials from CERT/CC outside of these fields. In addition, CERT/CC may make some of its collection available to the public in the near future: "we are currently working through some of those sensitivities so that we can release some samples publically and expect to do that within the next 6 months."[61]

Until such time, Stoner states that "we don't have a standard procedure" for providing samples to researchers. When requesting samples of malware that are not part of commercially available malware collections that CERT/CC subscribes to, "it's always a case by case decision that's dependent on the particular malware involved and any operational security concerns surrounding it."[62] This policy is in contrast to a library, archive, or museum, where allowing researchers access to materials would be a higher priority.

Computer security companies like F-Secure and Symantec keep repositories of malware for study, but Mikko Hypponen, Chief Research Officer at F-Secure, has said that no humanities researcher has ever requested access to F-Secure's collection.[63] Symantec Research Labs has developed Worldwide Intelligence Network Environment (WINE), a collection of high-quality datasets related to cybersecurity, including malware samples from 200 countries. Researchers can conduct experiments using WINE and analyze samples, but "raw data provided cannot be

---

[60] Ed Stoner, "RE: Malware Preservation Research," March 29, 2016.
[61] Ibid.
[62] Stoner, Ed. "RE: Malware Preservation Research," March 29, 2016.
[63] Mikko Hypponen, "Re: Malware Museum and Malware Preservation," April 7, 2016.

accessed anonymously or copied outside of Symantec's network."[64] WINE is only one of the many datasets and environments designed to be used by researchers in the computer security field, but these datasets often assume a high level of technical proficiency and focus more on quantity than a high degree of curation. Assessing and manipulating these large datasets may become an obstacle to researchers in the humanities, and while the scope of collecting for an institution like Symantec may be more comprehensive, a cultural heritage institution may have a sharper focus, for example, only collecting malware related to hacktivism.

Virus collector Cicatrix states that, in general, the "chances of getting virus samples from AV [antivirus] researchers / companies are slim. The agreed codes of conduct within the AV community generally preclude exchange of virus samples with someone outside this AV community."[65] Institutions like CERT/CC or Symantec have much higher barriers to entry for researchers than an archive, museum, or library. Antivirus companies or security research organizations may require multiple levels of approval to allow access, whereas, in a cultural heritage institution, a researcher may potentially gain immediate access.

While significant malware collections already exist, this thesis project has a slightly different focus: advocating for cultural heritage institutions to make malware a part of their permanent collections, and give it the same care that they would a collection of films, books, or personal papers. A cultural heritage institution would strive to collect robust metadata about particular pieces of malware and allow wide access it, especially if the institution is publically funded. While the antivirus and computer security research organizations have no responsibility

---

[64] Tudor Dumitras and Petros Efstathopoulos, "The Provenance of WINE" (Symantec Research Labs, n.d.), https://users.ece.cmu.edu/~tdumitra/public_documents/dumitras12wineprovenance.pdf.
[65] Cicatrix, "Collecting Computer Viruses: Fun or Folly?," March 1999, http://vxheaven.org/lib/static/vdat/epcolvir.htm.

to preserve historical malware several decades in the future, long-term preservation is the core mission of many cultural heritage institutions. Over time, CERT/CC, F-Secure, or Symantec may deaccession samples that do not serve their current purposes, but an archive, museum, or library intentionally collecting malware must consider the impact on long-term historical research before deaccessioning any items.

The Open Malware Project (previously Offensive Computing), managed by Danny Quist, provides researchers with straightforward access to malware samples. The project is affiliated with the Georgia Tech Information Security Center. Researchers can search the catalog using the hash value of a particular piece of malware or by the malware's common name. The project has existed since 2005 as a public source for malware samples. However, like the collection of CERT/CC and those of various antivirus companies, this site approaches malware collecting and cataloging from a computer security perspective and not a cultural heritage perspective. In addition, it remains to be seen whether long-term preservation will become an important part of the Open Malware Project's mandate.

Two Tracks of Discussion

The discussion that follows will chart two separate (but perhaps complementary) tracks related to malware preservation. The first track imagines a cultural heritage institution engaging in intentional and curated collecting of malware. The institution would identify historically important examples (of malware or related materials) and commit to preserving them long-term through a variety of strategies, which I discuss in detail in Chapter 5. The institution would also commit to making its collection accessible through public exhibition and other means.

The unique properties and risks of malware present challenges when an institution makes its collection accessible to researchers, especially if the researchers want to see malware demonstrated on a computer. I will outline both access challenges and strategies in Chapter 7. Institutions like the Computer History Museum in Mountain View, California seem well-positioned to take on such a collecting responsibility, especially since the museum has created a Center for Software History.

The second track involves examining the workflows of cultural heritage institutions who accession hard drives and disks into their collections and who may encounter malware. As Jane Gruning states, "Current digital archival practice often treats virus checking and quarantine as an unproblematic aspect of ingesting digital objects into an archival repository...that is often taken before any formal appraisal is done."[66] The digital preservation field stands to benefit from a discussion about the implications of malware quarantine and removal for items in archival collections.

After a brief review of malware's history (along with the sub-histories contained within) in Chapter 2, the exploration of the two tracks—intentional malware collecting and processing of infected born-digital artefacts—will occupy the remainder of this thesis. Chapter 3 will respond to poor analogies related to malware's status within born-digital collections. Discussion of best practices for handling malware-infected digital artefacts like hard drives and disks will take place in Chapter 4. This line of inquiry requires understanding what authenticity means for born-digital artefacts and whether the removal of malware when accessioning these items compromises the donation's authenticity or removes valuable information that a researcher may need in the future.

---

[66] Gruning, "Rethinking Viruses in the Archives."

Chapter 5 considers and assesses many different strategies for the preservation of malware.

Chapter 6 suggests how the removal of malware (if absolutely necessary) could be properly

documented. Chapter 7 will discuss the challenges in providing researchers access to malware.

Chapter 8 will cover strategies for assessing the risk of collecting malware. Finally, Chapter 9

will explore avenues for further research.

Chapter 2: A Brief History of Malware

Computer viruses and worms and similar incidents are not...antithetical to the general culture of networking and digitality but...at the very center of such enterprises. The digital virus is not solely an internal computer problem but a trace of cultural trends connected to consumer capitalism, digitality, and networking.
— Jussi Parikka, *Digital Contagions*

According to a 2013 report from Panda Security, 82,000 new malware threats are created per day, and a 2007 report by AV-TEST placed the number of unique malware samples (based on MD5 checksum) at 5,490,960 for that year.[67] Some malware threats may do major damage to computers and networks, or steal money or personal information, but others may be jokes, pranks, or a means to display creative payload screens.

In the current decade, "internet crime is one of the world's most profitable activities, as can be judged by the fact that a single zero-day iPhone exploit sells for $1M[million]."[68] In response to potential malware and hacking threats, the computer security industry has become a multi-billion-dollar operation. According to DazeInfo, "The global cyber security market is expected to grow from $106.32 billion in 2015 to $170.21 billion in 2020."[69] Clearly, both the creation of malware and the efforts to prevent it from infecting computers constitute a large sector of commercial and noncommercial activity in our current society.

Yet malware has a long history, and much of it involved experiments with software routines, utility programs, and artificial life. As Jussi Parikka demonstrates in *Digital*

---

[67] "Report: Average of 82,000 New Malware Threats per Day in 2013," *PCWorld*, accessed April 29, 2016, http://www.pcworld.com/article/2109210/report-average-of-82-000-new-malware-threats-per-day-in-2013.html.
[68] David S. H. Rosenthal, "Emulation & Virtualization as Preservation Strategies" (Andrew W. Mellon Foundation, 2015), https://mellon.org/media/filer_public/0c/3e/0c3eee7d-4166-4ba6-a767-6b42e6a1c2a7/rosenthal-emulation-2015.pdf, 23.
[69] Misra, Amit. "Antivirus Software Industry Growing, Despite Reports of Decline." *Dazeinfo*, August 25, 2015. http://dazeinfo.com/2015/08/25/antivirus-software-industry-growing-despite-reports-of-decline/.

Contagions, the development of computer viruses as we know them was historically contingent. Other uses could have existed for autonomous/"machinic" and self-replicating code (for example, a computer worm could have been developed by companies to collect overdue bills).

A historically representative collection of malware would help a researcher understand how viruses, and the discourse around them, developed from the 1950s to the present. In fact, what a "historically representative" collection of malware would consist of remains an open question. Is the malware that infected the most computers the highest priority to preserve? What about malware that used novel programming techniques? Or malware that was used in concert with offline political protest? To complicate matters further, many sites of "countermemory" exist at the margins of the dominant narratives concerning malware. This chapter will only suggest those particular sites and leave open pursuing these areas to future research.

Beginnings: 1950s–1970s

The early history of virus- and worm-like programs can be traced back to early debugging programs of the 1950s and 60s. These were self-replicating programs that functioned as useful (and non-malicious) utility loops. They behaved "according to instructions that made the programs copy themselves from one memory location to the next. This was intended to fill the memory space with a known value, consequently allowing it to be programmed with a new application."[70] These routines were often known as "rabbit programs"—a metaphor that underscored their ability to copy themselves rapidly. In addition, "writing self-reproducing programs with FORTRAN...was even considered to be form of popular entertainment in early

---

[70] Parikka, *Digital Contagions*, 40.

computer programming circles, an activity that was compared to playing video games."[71] The

computer games *Darwin* (1961) and its later adaptation, *Core War* (1984), had players program

"organisms" that resembled computer viruses and pit them against one another to see which

creature could take over the core memory of the computer.[72]
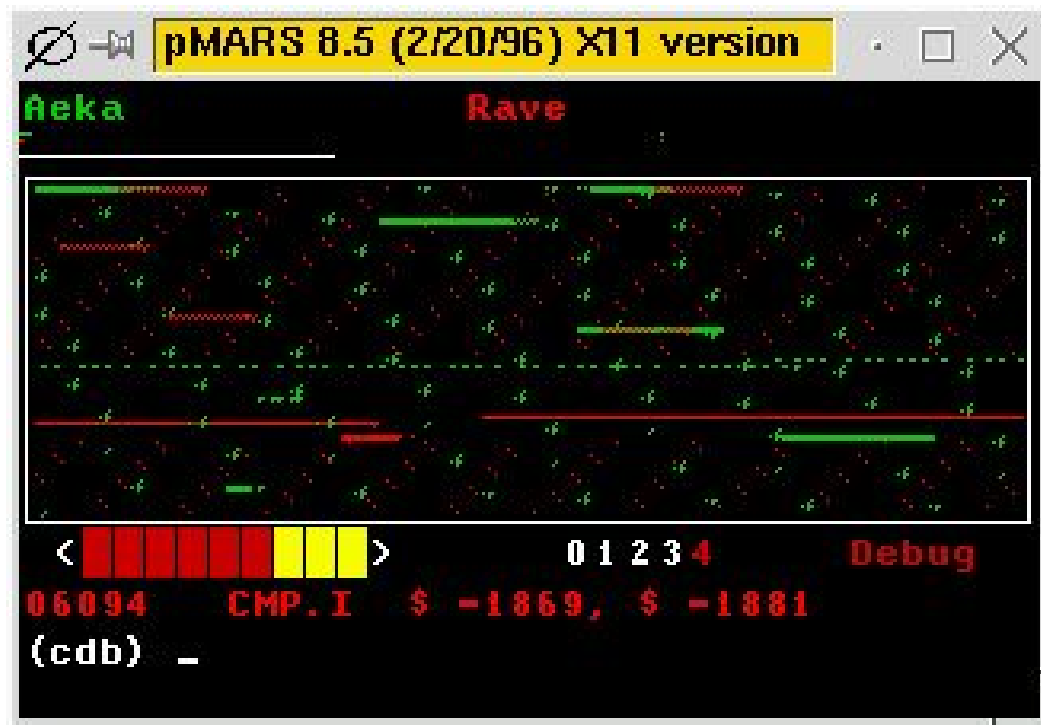
Figure 2.1: Screenshot of the game *Core War* running under a pMARS simulator. (Wikipedia)

In this early era,

> the majority...of computer viruses have ties to either the university or the corporation: the
> "Darwin" game (AT&T/Bell Labs, early 1960s), "Cookie Monster" (MIT, mid-1960s),
> "Creeper" and "Reaper" (BBN, early 1970s), "tapeworm" (Xerox PARC, early 1970s),
> and so on. Like early hacking activities, their intent was mostly exploratory. Unlike
> hacking, however, the language of biology quickly became a provocative tool for
> describing these encapsulations of code.[73]

---

[71] Ibid., 42.
[72] Ibid., 42–43.
[73] Galloway and Thacker, *The Exploit*, 83.

In the 1980s, computer scientist Fred Cohen would become noteworthy through his experimentation with computer viruses as artificial life. Cohen went on to coin the term "computer virus." and was also an early developer of antivirus software.

In 1971, Bob Thomas created the program Creeper to move between computers on ARPANET, an early computer network and the precursor to today's internet. Ray Tomlinson modified the program to make a copy of itself each time it moved to a new computer, thus creating the first computer worm.[74] While Creeper had no malicious payload, its onscreen message "I'm the creeper: catch me if you can" speaks to both the experimental and facetious attitudes of the worm's authors and the computer programming scene at the time. The companion program Reaper (1972), another computer worm written by Tomlinson, moved across ARPANET deleting copies of Creeper.

The late 1970s were an important period for the formation of "hacker culture," which saw its roots in several areas, including the Abbie Hoffman-influenced Yippie (Youth International Party) movement and the practices of phone phreaking—the independent experimentation with telephone systems by hobbyists who also developed exploits to make free phone calls. Fundamental to these social movements was "the idea that technology, particularly free use of phones, provided a centering mechanism for the movement as a whole, a technological infrastructure that members of the movement could access and usurp as their own." [75] Phreaker and hacker culture arose alongside social movements with strong critiques of authority and of consumer culture—movements that asked individuals to rethink their relationships to consumer technologies (such as computers or telephones).

---

[74] Parikka, *Digital Contagions*, 51.
[75] Thomas, *Hacker Culture*, 116.

One can also see these early roots reflected in later malware payload screens: the MS-DOS virus Frodo (aka 4K, released in 1989) was intended to display the message "Frodo Lives" on an infected computer's screen. This message, referencing Tolkien's hero Frodo Baggins, "was also a nod to a phrase made popular during the hippie era, reflecting the influence of 1960s counterculture on the nascent tech scene."[76]

The 1980s: A Media Sensation

The 1980s brought conceptual shifts in the philosophy of computer security as well as in the perception of experiments with self-replicating code. As network computing was not yet widespread, "in the mid-1980s computer security was defined in terms of access control and protection [to individual machines]."[77] A primary vector of malware was still an infected floppy disk. However, this emphasis on physical protection of equipment was about to change.

Malware began infecting significant numbers of computers in the 1980s—news reports introduced computer viruses and worms to the public for the first time. These media conceptualizations of viruses and worms displayed a rising tide of paranoia: "In both official administrative and fictional texts, viruses and worms were understood as remote agents of computer network intruders, articulated as *prostheses* of the criminal mind." [emphasis in original][78] Movies like *Tron* (1982) and *WarGames* (1983) depicted hostile computer systems as threats to the world outside computer networks.[79] Destructive metaphors of "breaking and entering" and "drunk driving" came to be associated with hacking and malware programming,

---

[76] Parikka, "Computer Viruses Deserve a Museum: They're an Art Form of Their Own."
[77] Parikka, *Digital Contagions*, 34.
[78] Ibid., 48.
[79] Ibid., 55.

and virus and worm events became "increasingly recognized by the legal system, the mass media and the computer (security) community."[80] The cover story for a September 26, 1988 issue of *Time* magazine focused on computer viruses, only a few weeks before the Morris Worm was released on November 2, 1988.

The computer virus epidemics of the 1980s were also indicative of a society now more dependent on computer systems. These large-scale infections were also a result of the proliferation of "plug in" personal computers that shielded users from the complexities of operation. On these consumer-friendly machines, which differed from earlier models that needed to be built and programmed from scratch, "a tiny piece of program code acting as a virus easily went unnoticed within the complex heart of automated computer operations."[81] Thousands of computers running MS-DOS, which had no integrated security, were a perfect platform for the spread of viruses.[82]

Brain (1986) was perhaps the first PC virus released to the public and spread through an infected boot sector of floppy diskettes. Two Pakistani software developers, Amjad Farooq Alvi and Basit Farooq Alvi, created the program and it was thought to be directed at software pirates, although others have theorized that it was a publicity stunt.[83] In a 2011 interview, the two brothers claimed to have created Brain to explore the flaws in the security of PCs running MS-DOS as well as to examine "how software moved around" through floppy disks.[84]

---

[80] Ibid., 54.
[81] Ibid., 58.
[82] This was stated in the proceedings of the Virus Bulletin conference, quoted in Parikka, *Digital Contagions*, 59.
[83] Parikka, *Digital Contagions*, 61.
[84] F-Secure. "Brain: Searching for the First PC Virus in Pakistan." *YouTube*, March 9, 2011. https://www.youtube.com/watch?v=lnedOWfPKT0.

The Morris Worm (1988) became the first individual piece of malware to make a big media splash. Using several strategies to propagate itself, including Sendmail, the Finger daemon, and password guessing, the worm spread to thousands of computers owned by universities and state officials.[85] Though the worm did not have a specific payload, it copied itself so many times on the same computer that it would cause the machine to crash. The worm multiplied so quickly that it slowed down sections of the early internet. The infection brought intense media coverage and Robert Morris, the worm's creator, was successfully prosecuted under the new Computer Fraud and Abuse Act for releasing the worm. He was sentenced to three years of probation, community service, and ordered to pay a fine of $10,050.

The problem for Morris, according to Parikka, was "not only what Morris did but where (and when) he did it." The incident acutely demonstrated the growing rift between enthusiasts experimenting with new technology and open access, on the one hand, and, on the other, a diverse world of users who had now become increasingly dependent on computer systems (for commerce and education among other uses) and did not necessarily share the same values.[86]

Issues of the hacker publication *2600* from the 1980s and 90s reveal a community divided over the release of the Morris Worm and the ethics of the creation and distribution of viruses and worms. They also reveal a group of people under siege by the authorities[87] (who may have had a "divide and conquer" strategy in mind similar to COINTELPRO's mandate to "disrupt, discredit, and misdirect" political organizations in the 1960s and 70s).

The outbreak of the WANK Worm (1989), which infected computers in the United States, Switzerland, and Japan, caught early security engineers off guard, as there was no

---

[85] Parikka, *Digital Contagions*, 80.
[86] Ibid., 83.
[87] See *2600* vol. 6, no. 4 (Winter 1989–90); vol. 6, no. 3 (Autumn 1989); vol. 9, no. 3 (Autumn 1992)

centralized communications or response team in the event of a worldwide malware incident.

WANK provided the impetus for the creation of the first international computer security alliance

called Forum of Incident Response and Security Teams (FIRST).[88]

During the 1980s, and even into the 1990s, among computing enthusiasts malware coding

often still had the status of a hobby, or perhaps a form of broadcasting or public address. Like the

Alvi brothers who created Brain, "for many, the motive was to follow how far their virus would

spread."[89] However, the late 80s marked a shift in conversations about computer security:

> As the security discourse moved from emphasizing *physical* safety [of computer systems]
> to securing the safety of *information patterns*, or internal security, it deterritorialized
> from practices based on human-to-human interaction to interactions between machines,
> programs, and the temporal, nonrepresentational processes of digital code...Security was
> usually designed to provide physical protection for data and equipment, but now access
> control and protection against manipulation were required at the level of software and
> network access.[90]

By the end of the 80s, viruses and worms had became understood as "enemies of the new

economy" within the discourse of the mainstream media.[91]

The cultural anxiety over computer viruses cannot be separated from the panic over the

HIV/AIDS epidemic that entered public consciousness in the 1980s: "some warned that

'[viruses] might do to computers what AIDS has done to sex', and computers had to have their

own prophylactics and guidance for safe use."[92] Such statements drew a connection between the

biological and the digital by comparing computer security and orderly computing to hygiene and

safe sex.[93] At least three computer viruses with AIDS in their name were created in the 1980s

---

[88] Dreyfus and Assange, *Underground*, 42.
[89] Quoted in Voon, "A Museum for the Blocky Graphics of Early Computer Viruses."
[90] Parikka, *Digital Contagions*, 48–49.
[91] Ibid., 89.
[92] Parikka, "Computer Viruses Deserve a Museum: They're an Art Form of Their Own."
[93] For more discussion on the connection between computer security and biological models of epidemic and illness see Parikka, *Digital Contagions*.

and 90s. Any researcher studying responses to the epidemic could examine this malware for clues about societal reactions to the spread of HIV/AIDS.

Figure 2.2: Payload screen of the AIDS MS-DOS virus (circa 1990). (Wikipedia)

The 1980s also saw malware created as conceptual art, such as the Rebel! Virus (1989): "In Italy, the pioneer of 'hacker art' Tommaso Tozzi [with the help of Andrea Ricci] designed a benign experimental virus...that displays the word 'Rebel!' on the computer monitor."[94] Artists' interest in creating novel pieces of malware or using malware code as a creative medium would continue into the present.

The 1990s: Going Viral

While Parikka writes that, "by the end of the 1980s and during the 1990s viruses were no longer simply toys for computer scientists and eager young hackers...but part of a multimillion

---

[94] Parikka, *Digital Contagions*, 75.

dollar business of computer crime and crime prevention,"[95] he slightly overstates the situation. While computer crime and the antivirus industry certainly got serious in the 1990s, a significant number of malware coders still created viruses without a specific criminal intent or clear economic imperative. The proliferation of non-destructive viruses with flashy payload screens continued well into the 1990s.

The increasing use of the internet by more individuals in post-industrial nations created an additional avenue for virus and worm propagation. However, in the early 1990s, "most viruses were still using the boot sectors of floppies to spread their code."[96] Second-generation viruses, introduced later in the 90s, were programmed to confuse antivirus software by "ballooning or pruning program code so that it always remains the same size."[97] Early antivirus software would often check the size of application files to detect viruses or worms.[98] Computers became more critical for everyday economic and social infrastructure during this decade and the mainstream discourse around malware coders pivoted from regarding them as "perverted" or "queer," but not necessary an imminent threat, to regarding them as simply "terrorists" by the 90s.[99]

In 1991, the release of *The Little Black Book of Computer Viruses*[100] raised a furor in the computer community. The book contained viral code and explanations about how to program viruses. Questions arose about the implications of publicly releasing malware code, including whether it was legal, whether it was permissible under the First Amendment, and ultimately

---

[95] Ibid., 86.
[96] Ibid., 88.
[97] Galloway and Thacker, *The Exploit*, 85.
[98] Ibid., 85.
[99] Parikka, *Digital Contagions*, 175–177.
[100] Ludwig, Mark A. *The Little Black Book of Computer Viruses*. Tucson, Ariz: American Eagle Publications, 1991.

whether it was ethical or responsible to widely disseminate knowledge that could be used for malicious ends. The debate over *The Little Black Book of Computer Viruses* signified the struggle between proponents of the dictum "information wants to be free" and those who took a more conservative approach, which continues to manifest itself in discussions about how to release "proofs of concept" that demonstrate security vulnerabilities and how much information should be publically released about contemporary malware.

The mid- to late-1990s saw the rise of a specific style of hacktivism that fused online and offline protest. While using computer hacking and malware for political purposes had always been a latent tendency, a new form of protest was brewing. More and more of the world's population was spending time online—a willing army of non-tech-savvy political activists were now connected to the internet.[101] Government agencies and corporations started to regard their websites as their public face. Resistance to corporate globalization schemes connected transnational constituencies of activists. The situation became a perfect storm that encouraged the blending of online and offline protest. Activists recognized that "the broad homogeneity of computer networks allows the virus to resonate far and wide with relative ease. Networks are, in this sense, a type of massive amplifier for action. Something small can turn into something big very easily."[102] The hallmarks of this style of hacktivism were displayed in 1994's "Intervasion of the UK" (resistance to the Criminal Justice Public Order Act of 1994), 1995's Italian "Net Strike" (against French nuclear policy), and 1996's "Chiapas Net Strike" (against the Mexican government's treatment of the Zapatistas).

---

[101] The first visual web browser, Mosaic, was released in 1993. Netscape Navigator followed in 1994. Both browsers are credited with popularizing the World Wide Web.

[102] Galloway and Thacker, *The Exploit*, 84.

In 1996's *Electronic Civil Disobedience,* the hacktivism group Critical Art Ensemble advocated abandoning street protest for cyberspace: "resistance—like power—must withdraw from the street. Cyberspace as a location and apparatus for resistance has yet to be realized. Now is the time to bring a new model of resistant practice into action."[103] In the late 1990s, a group called the Electronic Disturbance Theater released the Java applet FloodNet (which it called "an example of conceptual net.art").[104] FloodNet allowed participants to engage in a "virtual sit-in" by continually reloading the target website's homepage with their computers—in effect, becoming a voluntary part of a botnet. The computer security community perceives this kind of activity as a malicious distributed denial-of-service attack, as enough web traffic can take a site down (similar to how protesters can blockade the entryway to a building during a sit-in). FloodNet's users were also able to leave customized messages on the error log of the server that they targeted. The targets included Mexican government websites and the website of the Mexican stock market as well as the sites of various financial institutions.

---

[103] Critical Art Ensemble, "Electronic Civil Disobedience," 1994, http://www.critical-art.net/books/ecd/ecd2.pdf, 25.
[104] Brett Stalbaum, "The Zapatista Tactical FloodNet," *Electronic Civil Disobedience*, accessed May 1, 2016, http://www.thing.net/~rdom/ecd/ZapTact.html.
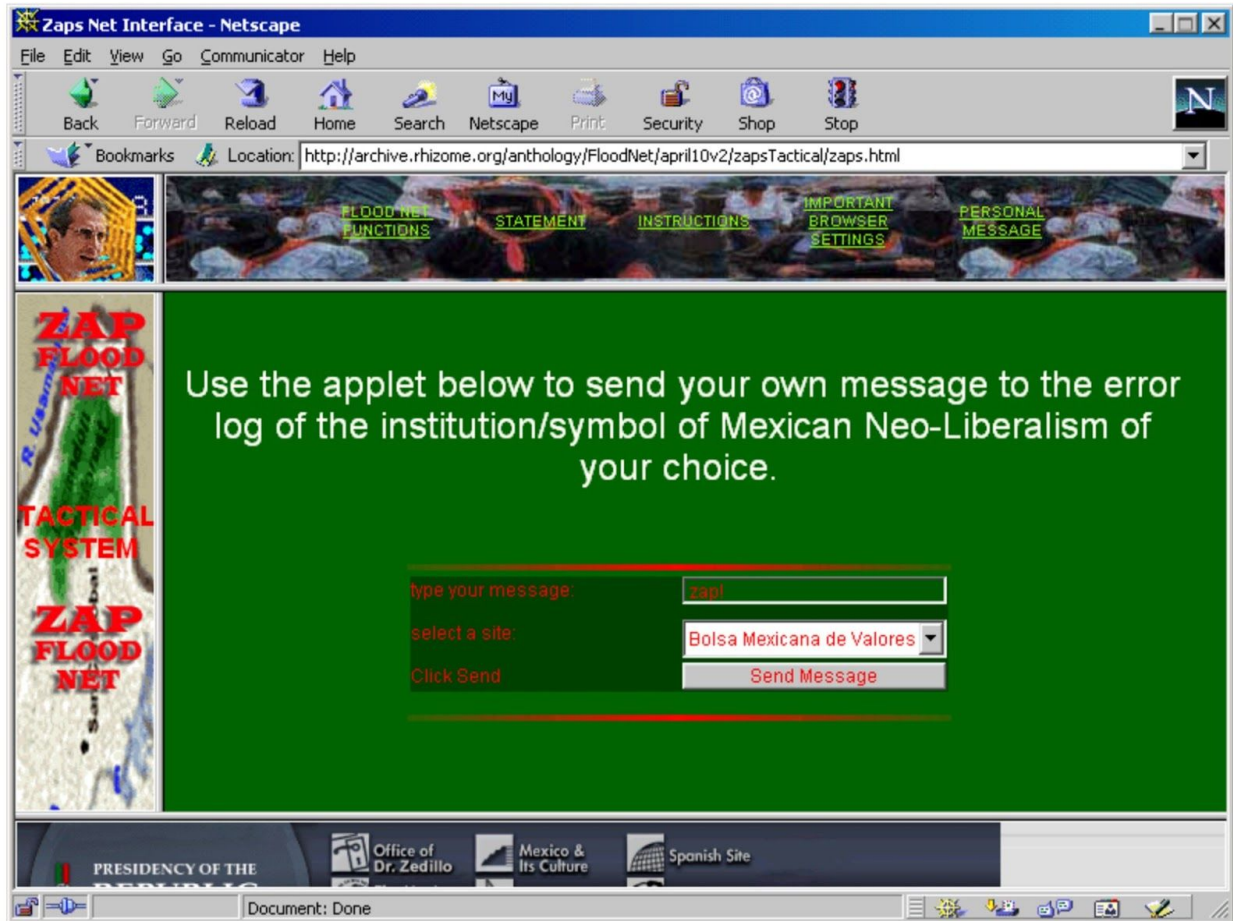
Figure 2.3: Screenshot of Floodnet recreated through emulation. Through the applet, the user was able to leave their own customized error messages on the servers of target institutions. (Rhizome Net Art Anthology)

Throughout the 1990s, Wintel machines (computers with Intel-manufactured CPUs running Microsoft Windows) dominated the marketplace for personal computers. Homogeneity of machines became a boon for malware developers: "computer viruses thrive in environments that have low levels of diversity…Viruses and worms exploit holes and in this sense are a good index for oppositional network practices."[105] Software that created backdoors, such as Back Orifice (1998), and viruses such as Melissa (1999) and ILOVEYOU (2000), exploited

---

[105] Galloway and Thacker, *The Exploit*, 84.

vulnerabilities in the Microsoft Windows operating system, the email program Microsoft Outlook, and the Microsoft Office suite in order to replicate themselves or take control of computers. One could argue that these pieces of malware, especially Back Orifice, made a public statement, challenging Microsoft's shortsighted policies on security and can be seen as aggressive "proofs of concept."
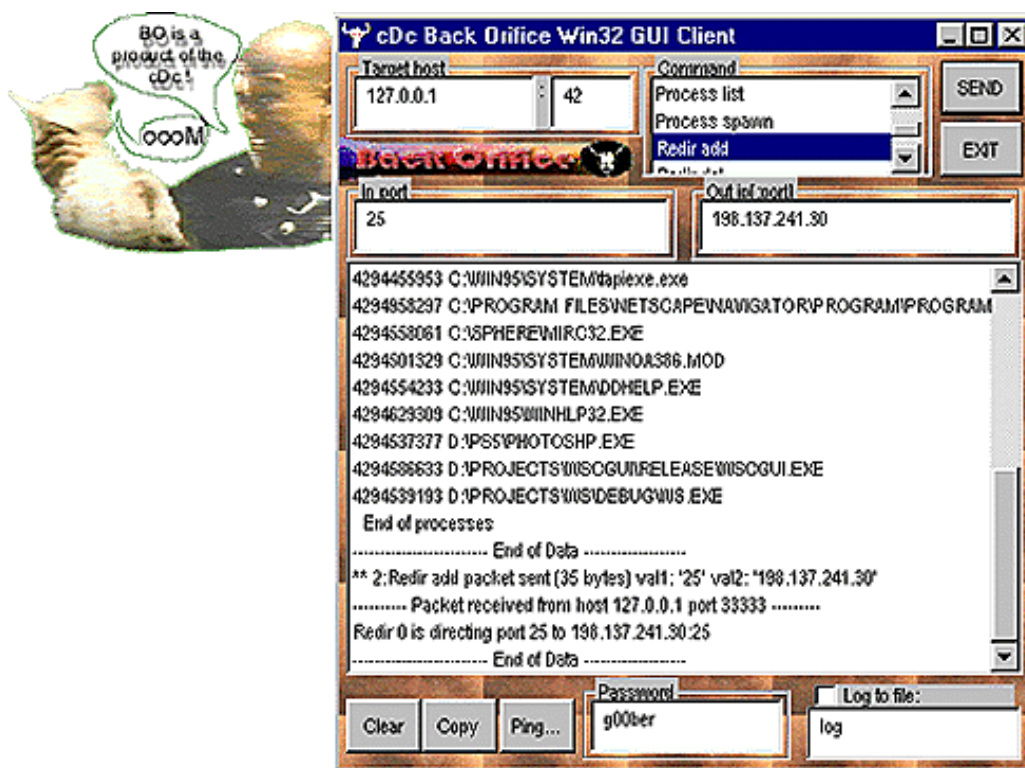
Figure 2.4: A screenshot of the Back Orifice client window. This interface could be used to remotely control an infected computer. (Cult of the Dead Cow)

The aptly named Concept (1995) was the first Microsoft Word macro virus. The rise of macro viruses meant that malware code was no longer limited to executable files—"even data files could now be a threat, and defences had to adjust accordingly."[106] Email became another

---

[106] John Aycock, "Stux in a Rut: Why Stuxnet Is Boring," *Virus Bulletin*, September 1, 2011, https://www.virusbulletin.com/virusbulletin/2011/09/stux-rut-why-stuxnet-boring.

malware vector and Happy99 (1999) was the first email worm that could send itself to others.

Melissa (1999) was both an email and a Word macro virus that proved highly virulent and "was

said to have infected up to 20 percent of computers worldwide."[107] Many pieces of malware from

this era disguised themselves as common file types or email attachments. Using malware as

evidence can provide historical perspective on the kinds of files individuals believed were

important to open. Analyzing the content of this malware could also shed light on their cultural

values as well as their level of trust regarding computer technology. For example, ILOVEYOU,

which claimed to contain a love letter, exploited an individual's desire to see an amorous

message from a person they knew.

The 1990s also saw the rise of the polymorphic virus. Polymorphic viruses can evade the

typical antivirus heuristic of detecting "signatures"[108] of malware. These viruses "are able to

change themselves at the same time that they replicate and distribute themselves. In this case,

computer viruses are defined by their ability to change their signature and yet maintain a

continuity of operations (e.g., overwriting code, infiltrating as fake programs, etc.)."[109] The rise

of polymorphic viruses required the radical retooling of the way antivirus software worked.[110]

Subsequently, third generation viruses could "intercept and mimic the antivirus software,

thereby performing fake file scans." Fourth-generation viruses "employ 'junk code' and 'attack

code' to carry out multi-pronged infiltrations, in effect overwhelming the computer's antivirus

software."[111] The conflict "between morphing viruses and antivirus discourse developed into an

---

[107] "Top Ten Most Destructive Computer Viruses of All Time," *Crunkish*, accessed May 1, 2016, http://crunkish.com/top-ten-worst-computer-viruses/.
[108] A signature is a byte pattern that is unique to a piece of malware. Antivirus software and signatures will be discussed in more detail in Chapter 6.
[109] Galloway and Thacker, *The Exploit*, 85.
[110] Aycock, "Stux in a Rut: Why Stuxnet Is Boring."
[111] Galloway and Thacker, *The Exploit*, 85.

information commodity spiral" with an "arms race" of the antivirus industry trying to detect and

thwart more sophisticated viruses and virus coders trying to outwit antivirus software.[112] Thus,

the emergence of computer viruses and societal reaction to them has demonstrated that "digital

capitalist culture also seems to be the first system that has really succeeded in converting its own

accidents to its own profit."[113]

The 2000s–The Present: Malware Goes Commercial

During the first decade of the 2000s, malware underwent a significant shift in its

status—it became almost entirely commercial. Fizzer (2003) was perhaps the first piece of

malware created for the sole purpose of making money; "it would infect computers, build a

proxy network out of them, so you could reroute proxy or email traffic through them, and that

service was then sold to spammers."[114] This "business model" of creating avenues for sending

spam and then selling them persists in the malware of the present day. In addition, corporations

can now hire hacking companies who may use malware to go after rivals and take down their

computer systems. Hacking Team and Equation Group are two such firms which can be hired to

develop malware and launch attacks. Malware coders in the current decade are now, more than

ever, driven by financial gain—stealing credit cards numbers, breaking into financial institutions,

or holding individual or company data for ransom. Mikko Hypponen suggests that "the source of

---

[112] Parikka, *Digital Contagions*, 99.
[113] Ibid., 100.
[114] "The History and the Evolution of Computer Viruses: 2003-2008," *Privacy PC*, March 25, 2012,
http://privacy-pc.com/articles/the-history-and-the-evolution-of-computer-viruses-2003-2008.html.

malware today is 99 percent criminal gangs...there are organized criminal gangs, making millions from their attacks."[115]

Increasingly, governments have developed or purchased malware, especially spyware, heralding an era of increasingly powerful cyberweapons and potential cyberwar. The Stuxnet (2009) computer worm, believed to be jointly developed by the United States and Israel, infected programmable logic controllers to sabotage nuclear centrifuges in Iran. Hypponen believes that Stuxnet was the end result of the 2008 Comprehensive National Cybersecurity Initiative signed by then-president George W. Bush.[116]

The current decade has also seen the rise of subclasses of malware called ransomware and scareware, which often have connections to organized crime. Scareware uses tactics such as pop-up ads on webpages to dupe users into downloading or paying for software that is basically useless, or worse, contains malware like viruses or worms. The scareware will often advertise itself as being antivirus software or another kind of utility software.

Ransomware allows attackers to remotely lock a victim out of their own computer, or restrict the victim's access to their files, often through encryption. The program then demands a ransom from the user to restore access. Users can become infected with ransomware by opening an email attachment, or by clicking on a link to an attack website within an email, web page, or pop-up ad. Ransomware called Cryptolocker (2014) encrypts files on a hard drive and refuses to decrypt them until the ransom is paid, but other ransomware may simply lock keyboard and mouse input. Reveton (2012) pretends to be locking a user's computer on behalf of the United States Department of Justice and states that the user has images of child pornography on their

---

[115] Quoted in Mat Honan, "Why Hackers Write Computer Viruses," *Gizmodo*, August 4, 2011, http://gizmodo.com/5827405/why-hackers-write-computer-viruses.
[116] Ibid.

computer, then it demands a fine to restore access. The connection to the Department of Justice

is simply a ploy to get people to pay.

Collecting payments through ransomware has become a cottage industry in Russia and

Ukraine. Crime rings have set up offices similiar to call centers where agents monitor

ransomware payments and unlock computers.[117] This recent phenomenon, and the malware that

supports it, should be of interest to those studying the history or culture of organized crime.

Attacks by criminal groups may even continue in the offline world as the son of Eugene

Kaspersky (head of antivirus company Kaspersky Labs) was kidnapped in Moscow in 2011 and

some suspect a revenge motivation.[118]

While earlier malware announced itself to the user with creative payload screens,

contemporary malware creation is less focused on appearance and is often more clandestine in its

infection, using complex methods to avoid detection.[119] Malware coders continually employ

more sophisticated forms of obfuscation that go beyond polymorphism. For example, some

viruses reside at the kernel level of a computer in order to hide from antivirus software.[120]

Cyberweapons like Stuxnet attempt to make their effects seem like the result of an accident.

Contemporary malware coders launch "zero-day attacks," where vulnerabilities are exploited so

quickly that security companies and software companies do not have time to respond to them or

patch them.

Botnets have also became more common. John Aycock argues that "the necessary

condition was the appearance of a large pool of vulnerable, always-on, always-connected

---

[117] "Darkode," Podcast, *Radiolab*, September 21, 2015, http://www.radiolab.org/story/darkode/.
[118] Honan, "Why Hackers Write Computer Viruses."
[119] Quoted in Voon, "A Museum for the Blocky Graphics of Early Computer Viruses."
[120] See Nicole Perlroth, "Researchers Track Tricky Payment Theft Scheme," *New York Times*, November 24, 2015, http://bits.blogs.nytimes.com/2015/11/24/researchers-track-tricky-payment-theft-scheme/?_r=0.

computers. These computers have been repurposed by adversaries, unbeknownst to their owners, for stealing information, sending spam and conducting distributed denials of service."[121] Botnets presented a new challenge for the computer security industry, namely that security analysts had to "look beyond a single computer, beyond a single network, and beyond a single country."[122]

Motivations for Malware and Hacking

Much malware, at least until the 2000s, presents itself as a joke or a prank perpetrated by a clever or skilled coder. Douglas Thomas draws from a rich theoretical literature on pranks and their relationship to the social fabric to discuss a specific style of hacking, that of the prank that disturbs authority, which takes place on the hackers' own turf—computer and information systems. Thomas argues that:

> technology is exploitable primarily because of cultural attitudes toward it. Even while people are distrustful of technology or suspicious of it, they cede authority to those who control or appear to control it. The hacker, who is able to master technology, speaks with two voices—the voice of adult authority, with which he asserts control, and the voice of boy culture with which he resists and assaults the values and norms of the adult world. Technology, like the figure of the hacker, is thus rendered undecidable, caught between two discourses, one of mastery and one of subversion.[123]

In this manner, malware exploits the majority of the population's social relationship to technology.

Anthony Rotundo argues that pranks are "skirmishes in a kind of guerilla warfare that little boys wage against the adult world."[124] While a certain kind of hacking and pranking may have some connection to attitudes associated with "boy culture," it would be a mistake to

---

[121] Aycock, "Stux in a Rut: Why Stuxnet Is Boring."
[122] Ibid.
[123] Thomas, *Hacker Culture*, 48.
[124] Quoted in Thomas, *Hacker Culture*, 47.

discount hackers and malware creators who are not male, and are often invisible in conventional

narratives. A virus coder named Gigabyte "wrote her Sharpei worm with Microsoft's C#

programming language apparently to teach a lesson to sexists who think women cannot code.

According to the 17-year-old Belgian, virus writers are not merely pimple-faced male teenagers,

not merely the faces constantly repeated and circulated in media representations."[125] Gigabyte's

actions also point to the fact that the motivations for the creation of malware always exceed

conventional expectations. Dark Avenger, a virus writer from Bulgaria, intentionally launched

hacking attacks on virus researcher Vesselin Bontchev. However, the true identity of Dark

Avenger remains mysterious and some have speculated that Dark Avenger and Bontchev were

promoting each other or are the same person.[126]

Finn Brunton conceives of spammers and other digital deviants as operating in an

emerging frontier in the constitution of our collective awareness:

> Spam persists and diversifies because we are living through a major complex transition in the constitution and management of our own attention, a transition moving faster than our governance, our metaphors, and our software can keep up with. Spammers—the disbarred, lawyers, impoverished con artists, would-be pornographers, credit card thieves, and malware coders—are the avant-garde, the wildcatting exploiters of this transition. They find domains where salience is being generated, whether in a comment thread, a search engine result, a social media platform, or your email inbox, and move to commandeer it...In their crude way, they show the rest of the online population the network's new capabilities, the new forms of attention and community experience, which we have not yet fully understood.[127]

Malware coders and hackers alike can surprise others in demonstrating what computers and

networks are capable of. Media theorist McKenzie Wark identifies the ontological status of a

hack as the exploitation of "virtuality." He writes, "To the hacker, what is represented as being

---

[125] Parikka, *Digital Contagions*, 145.
[126] See Gordon, Sarah. "Inside the Mind of Dark Avenger." *VX Heavens*, January 1993.
https://download.adamas.ai/dlbase/Stuff/VX%20Heavens%20Library/static/vdat/ivdarkav.htm.
[127] Brunton, *Spam*, 197–198.

real is always partial, limited, perhaps even false...there is always a surplus of possibility expressed in what is actual, the surplus of the virtual...To hack is to release the virtual into the actual."[128] Wark focuses on the ingenuity of the hacker (or malware coder), using his or her imagination to reconfigure what others would simply pass over as a "matter of course" or as a limit of the system. Malware coding explores its own surplus and reconfiguration of how computer systems can be used: to spread a personal or political message, to sabotage other computer systems, or to create an army of zombie computers. Such reconfigurations ought to merit the attention of cultural critics and even would-be revolutionaries.

Malware as Fine Art

    coding a virus can be creative —Payload screen of the Spanska.1500 Virus (1997)

    There are multiple potential perspectives for interpreting malware programming's place within the history of art. The Biennale.py virus, created by Eva and Franco Mattes and the hacker group epidemiC, became an experiment in viral performance. Biennale.py was released at the 2001 Venice Biennale and the media hype surrounding its release was at least as, if not more, important than the virus's creation. The artists had informed antivirus companies about the virus, and in a press release stated that Biennale.py would be "a form of global counterpower."[129] The artists' hyperbolic rhetoric and the cloud of mystery surrounding the virus (Would Biennale.py damage computers? Was it even legal to release?) created a tense atmosphere at the Biennale, though it's unclear how many computers, other than the ones exhibited, were infected. The virus

---

[128] McKenzie Wark, *A Hacker Manifesto* (Cambridge, MA: Harvard University Press, 2004), 74.
[129] epidemiC, "Biennale.py," accessed May 1, 2016, http://epidemic.ws/biennale_press/01.htm.

had no malicious payload and contained removal instructions within its code. In addition to

releasing the virus, the artists sold t-shirts, posters, and CD-ROMs with the virus's code at the

Biennale. Their concept was to spread viral code not just through computers, but through

material objects and art markets.

Malware as fine art is not just about creating media spectacles: Hellraiser, an influential

virus writer, has said that "viruses are an electronic form of graffiti."[130] The statement is

especially significant when one considers that graffiti was regarded as vandalism long before

certain strains of it were acknowledged as "street art." In addition,

> many net and software art projects dealing with viruses have attempted to debate digital
> security, and in many cases asked how malware is related to issues of privacy and
> control. Hacker-artist Luca Lampo, for example, has suggested that the fear of computer
> viruses and other "monsters" of digital culture was part of a longer history of projected
> (Western) fears, replacing previous monsters such as Soviet Russia.[131]

Art that incorporates malware can also be viewed as part of the history of participatory

aesthetics, for example the software FloodNet.

Some malware produced as fine art appears influenced by previous experiments with the

cultivation of chance and noise. By using malware, these artists are exploiting certain

affordances generated by the system itself—in other words, the "noise" that computer networks

generate is malware. A virus is not the "Other of the system...Instead, the thematics of noise are

relational and part of the establishment of systems. Noise is to be understood as a differentiation

of a system, in other words, a system creates its noise and viruses...The virus, the noise, is the

bastard offspring: unrecognized yet not foreign."[132] Noise is inherent to the system, and although

---

[130] Quoted in Alessandro Ludovico, "Virus Charms and Self-Creating Codes" (digitalcraft, n.d.),
http://www.digitalcraft.org/iloveyou/catalogue_alessandro_ludovico_virus_charms.htm.

[131] Parikka, "Computer Viruses Deserve a Museum: They're an Art Form of Their Own."

[132] Parikka, *Digital Contagions*, 38.

considered by some to be undesirable, to others it becomes raw material for creativity. "Noise does not automatically mean chaos. In other words, viruses and similar types of programs do follow certain types of logical algorithmic patterns and are products of a 'rational' piece of code."[133]

James Hoff inserts malware code into the existing code of digital paintings and audio files to generate glitches. In addition, "Artists such as Joseph Nechvatal incorporated viral code into new forms of digital painting to infect and break down the images produced. Associated avant-garde art techniques of randomness and variation became part of digital visual culture."[134]

Minoritarian History

Histories of malware are among the marginal histories of computing lying in wait to be unearthed. Those who study the history of media can "aim to follow the detours and the experiments that remain virtual, yet real, in the shadows of the actuality of hegemonic understanding."[135] Parikka sees work on the media archaeology of computer viruses as continuing along the lines of such "countermemory" projects as new histories of women, children, LGBTQ people, and people with disabilities. I support his call for the "dislodging of established points of subjectivity from their places, opening up new sites and territories of acting and remembering."[136] Could a malware archive open up new ways of remembering, or of writing the history of digital culture? Perhaps building a collection of malware supports the

---

[133] Ibid., 38.
[134] Parikka, "Computer Viruses Deserve a Museum: They're an Art Form of Their Own."
[135] Parikka, *Digital Contagions*, 24.
[136] Ibid., 25.

reconceptualization of the history of information technology, or provides evidence for a minoritarian history of computing.

The Digital World as "Third Nature"

Georg Wilhelm Friedrich Hegel, Karl Marx, and György Lukács theorized the existence of "second nature" as the human social sphere of institutions, such as governments and economic systems, as well as the interior world of individuals. McKenzie Wark proposed that "perhaps the digital culture of the late twentieth century is to be understood as nature, a *third* nature that supplements the two previous ones."[137] If the digital world constitutes "third nature," then malware represents a segment of the ecology of that digital world (almost in the same way we regard biological viruses within the biosphere). This point may explain why "although there have been constant attempts to pin down viruses and worms as projections of human nature—whether in the form of malicious and frustrated hackers or of misguided teenagers, and so forth—they continue to be treated as quasi-natural entities."[138] If the digital world is third nature, preserving malware would be akin to the conservation of particular ecosystems. The media ecology in which malware is created and released constitutes a dynamic network of human beings, media, and technology, an environment "as imperceptible as water to a fish."[139] Positing a "third nature" has resonant impacts on our own views of ourselves as human beings in an increasingly complex automated world. Parikka argues that in understanding malware and network culture, we can reinterpret our own notions of agency and subjectivity: "The subjectivities of...network culture

---

[137] Quoted in Parikka, *Digital Contagions*, 9.
[138] Ibid., 10.
[139] Ibid., 17.

are increasingly nonhuman, which further underlines the need for novel conceptualizations of culture, agency, and subjectivity."[140]

---

[140] Ibid., 14.

Chapter 3: A Series of Inaccurate Analogies

In my research, I encountered several criticisms of both the intentional collection of malware by cultural heritage institutions and the preservation of malware-infected versions of digital artefacts. These critics have attempted to draw analogies between malware infection and issues that are already well-understood in the treatment and care of archival collections. I will examine each of these analogies to help clarify the debate and elucidate how malware fits within the collecting mandate of archives, museums, and libraries.

Dust, Mold, or "Digital Dirt"[141]

Malware infecting a hard drive donated to an archive has been compared to mold, dust, or other kinds of contaminants that must be cleaned from items like photographs or reels of film. To archivists as well as the original user of the donated computer, malware may be unwanted, like dirt or blight. However, mold, dust, and other kinds of environmental degradation are ultimately different from a malware infection. Dust and mold are not produced through direct human intervention and therefore are not part of the human cultural record. This is not the case for a malware infection.

However abstracted through computer technology (and however undesirable) a malware infection is essentially a person-to-person interaction and the malware's code is "a product of the human intellect, resulting from our present day culture."[142] Ultimately, the beginning of a virus's

---

[141] This term is borrowed from Jussi Parikka's book *Digital Contagions*.
[142] Franziska Nori, "I Love You" (digitalcraft, 2002), http://www.digitalcraft.org/?artikel_id=284.

life is based on the intentional decision of a human being to create or distribute it. The same cannot necessarily be said for the mold on a reel of film.

Information that reveals past relationships between individuals may be contained within malware, for example the motivations of those who wrote it, how the computer was infected, or perhaps how the infection may have affected the experience of the computer's user. None of this kind of information is contained within mold growth or other kinds of physical or environmental contamination. While an artist or writer may have tried to avoid getting malware on their computer, once infected, the malware becomes part of the environment that the individual must deal with. If a cultural heritage institution aims to reflect the user experience of this person, it must at the very least acknowledge the effects of the malware.

Computer viruses are not just trash or "digital dirt." Jussi Parikka argues that "they reveal characteristics of a specific digital ecology and actually can provide an essential viewpoint on our network culture."[143] A human being creates and releases a virus in order to communicate with others, whether or not one agrees with the motivations of the malware coder: "Computer viruses have not been mere 'accidents of nature,' but are seen also as cunning forms of vandalism and an indication of sabotage mentality...Virus writers, whether or not they have targeted specific companies or individuals, must know that their programs, once unleashed will soon become *uncontrollable.* [emphasis in original]"[144] The vastness and complexity of contemporary computer networks make malware a form of disorderly mass communication. An

---

[143] Parikka, *Digital Contagions*, 215.
[144] Ibid., 34.

individual malware infection is simply one record of a communication or interpersonal transaction. [145]

Furthermore, institutions generally keep very detailed records on mold infection and cleaning. If it becomes necessary for an institution to remove malware from its collection items, the removal should get a similar amount of documentation. If an institution's Information Technology department handles malware removal, documentation about the removal should be made available and understandable to both archivists and researchers.

Broken Cassette Housing

A malware-infected hard drive has been compared to the damaged housing of a video cassette. Since cassette housings for a given video format are functionally the same, a broken housing can be swapped out for a new one with no loss of information, provided any existing annotations on it are transferred or documented. This analogy implies that malware's provenance information for an item is superfluous and, if the artefact is a hard drive, that preserving the infection is not an important part of the original working environment. As I have discussed, particularly for a political activist's computer infected with spyware, useful information may exist for a future researcher only when considering the computer's entire working environment as a repository of evidence. [146] In this case, removing malware from an infected hard drive would be similar to trashing a broken cassette case that had FBI notations written on it without first

---

[145] One might argue that if the malware was created through a program that generates malware automatically (such as a Virus Creation Laboratory) the malware's creation is not a form of interpersonal communication. Even if the virus was created through software (and the individual responsible did not write a line of code), this person still made the intentional decision to create the virus and release it into the world.

[146] It is especially important to consider the entire computer system, potentially including the hardware. As I discussed in Chapter 2, malware can hide in obscure sectors of computer systems.

recording those notations. Discovering and documenting spyware on infected hard drives gives

future researchers the ability to know whether an activist had been surveilled without needing to

interface with government bureaucracy (for example, by filing a Freedom of Information Act

request), which decides whether the information is appropriate to release.

In other cases, such as with an artist, removing malware could be akin to disposing of a

cassette housing that an artist had customized with ink drawings or paint, or disposing of

documentation of the artist's creative process.

Dynamite or Nitroglycerin

Unlike explosives in the offline world like dynamite or nitroglycerin, malware does not

just "blow up" computers. Each piece of malware has specific targeted effects as well as

limitations. Some have time delays to a specific date, such as the Michelangelo Virus (1991),

which is set to release its payload on the birthday of the artist Michelangelo, and the Frodo

Virus, which releases its payload on the birthday of Bilbo Baggins. Viruses and worms can also

contain limitations on how their payloads are released. The WANK Worm's creator(s)

programmed it not to infect computers in New Zealand, which became a significant piece of

information in understanding both the motivations of the worm's creator(s) and the worm's

origin. In fact, some malware very intentionally has no "explosive" effect at all. For example, if

its creators wanted to bring attention to an operating system vulnerability (or to themselves), but

did not want to actually harm computers. A live demonstration of malware may show a

researcher something more than just a dramatic deletion of files or the wiping of a hard drive.

Different malware works in very different ways and the differences in its methods of infection,

the specific vulnerabilities it exploits, and the assumptions that it makes (for example, the Michelangelo Virus assumes that a computer it infects has a very specific hard drive geometry) are significant.[147]

What Malware Infections Can Reveal

In addition to the previous suggestions, malware on a computer could be evidence of visiting particular websites or downloading particular files. There is also the possibility that the presence of a malware infection could indicate the provenance or source of the donated disk or hard drive. Through understanding where certain malware originated or how it spread, a researcher could potentially trace where a disk was written, especially if the malware is an extremely rare variant. An analogy can be drawn between this kind of forensic analysis and researchers who trace the origin of paintings through analyzing the chemical composition of the paint itself.

For institutions like universities or corporations, malware infections could be evidence of several things, all of which may have historical significance. Malware-infected computers belonging to a CEO or high-level executive could be evidence of corporate espionage by other companies. It could also be evidence of competitors contracting hacking groups to attack the company if the malware in question is traced to a specific group. Malware on a corporation's computers could also be part of the story of grassroots resistance or a political movement against the company. Computers infected with cyberweapons, such as Stuxnet, may be evidence of a covert cyberwar campaign.

---

[147] "Michelangelo (computer Virus)," *Wikipedia*, accessed May 1, 2016, https://en.wikipedia.org/wiki/Michelangelo_(computer_virus).

Finally, saving malware on a hard drive or disk could retain evidence of the history of malware collecting itself. There is no shortage of amateur and professional malware hunters and collectors on the internet. Keeping collections intact could show how malware collecting developed as a hobby and professional practice, including which individuals were collecting in different eras, how they stored their collections, their collecting policies, and what (if any) safety precautions they took.

Chapter 4: A Gap in Institutional Practice

Born-digital physical storage media like hard drives, floppy disks, and optical discs have a limited lifespan and will not remain readable forever. Thus, the media preservation community has reached a consensus that creating a disk image (an exact copy of all of the data on a hard drive or disk as a digital file) of the physical media and seeking to preserve this disk image (by storing redundant copies of the image in different places and on different storage media) is the best course of action.

To this end, archivists have adopted some of the tools and principles of digital forensics. Digital forensics tools and workflows were developed by the law enforcement community as a method to collect digital information that is permissible as evidence in a court of law. Practitioners of these methods recognize what is a basic principle of forensic science—"Every contact leaves a trace." Furthermore, they understand that the dictum "is more, not less true in the delicate reaches of computer systems."[148]

Digital forensics practioners must be able to conclusively prove that the data contained within a disk image exactly matches that of the original hard drive or disk recovered as evidence. Therefore,

> when a hard disk is duplicated for forensic investigation it is not enough to simply copy the files in the usual manner...Instead, an investigator will want to create a so-called bitstream image [or disk image] of the original file system. A bitstream is exactly that: every bit recorded on some original, physical instance of storage media transferred in linear sequence to the copied image...This means that all of the other ambient data on the original media is retained as part of the forensic object, including even...data in 'bad' or corrupted sectors no longer otherwise accessible.[149]

---

[148] Kirschenbaum, Matthew G. *Mechanisms: New Media and the Forensic Imagination*. Cambridge, Mass.: MIT Press, 2008, 49.
[149] Ibid., 53.

Archivists and conservators have modified these practices to suit the needs of accessioning born-digital artefacts, like hard drives and floppy disks, into their collections. However, when examining these kinds of physical media, archives, museums, and libraries sometimes encounter malware.

Jane Gruning addressed some of the issues that arise when cultural heritage institutions encounter malware in her poster on "Rethinking Viruses in the Archives." Gruning observed that:

> current digital archival practice often treats virus checking and quarantine as an unproblematic aspect of ingesting digital objects into an archival repository; it is simply a step in the process, that is often taken before any formal appraisal is done. Viruses are separated from the records, and then forgotten about or perhaps disposed of.[150]

While virus scanning is important to ensure the integrity of files in a digital repository, there is a noticeable lack of discussion on how to handle malware infections in the archival profession. If archivists remove malware from digital artefacts within their collections they risk "creating a gap in the history of computers and their use in our society – a gap that we could potentially avoid...Archivists need to rethink how we, as a profession, are addressing this issue."[151]

In order to survey current workflows for processing born-digital artefacts, I spoke with archivists from several institutions—New York University Libraries, The Museum of Modern Art, and Johns Hopkins University. All of the individuals I spoke with use digital forensics workflows and tools for creating disk images at their respective institutions.

---

[150] Gruning, Jane. "Rethinking Viruses in the Archives." Poster presented at the Archival Education and Research Institute, 2012. https://www.ischool.utexas.edu/~janegru/images/Gruning_AERI2012.pdf.
Given the responses I received from interviews with archivists and conservators, I am not sure if I agree with Gruning's assessment of how archives are handling malware. However, I have placed her thoughts here as a potential warning to institutions considering removing or discarding malware.
[151] Gruning, "Rethinking Viruses in the Archives."

Cultural heritage institutions are still in the process of developing standardized workflows for imaging hard drives and disks and ingesting disk images into their digital repositories. Workflows for handling born-digital artefacts are still evolving and new best practices are emerging. Many institutions, such as the Bancroft Library at the University of California Berkeley, have yet to encounter malware in the born-digital collections they have processed.

The archivists and conservators I spoke with responded to questions about how to handle malware infections with answers like "the situation hasn't come up yet," "we don't have a standard procedure," or "we haven't made a decision on this." At present, decisions about handling malware-infected digital artefacts are mostly ad hoc and the need for a more thorough discussion about workflows is apparent.

Despite Gruning's concerns, all of the staff I spoke with had reservations about completely discarding or removing any malware, particularly without any documentation. Instead, they adopted a "wait and see" approach, either waiting to fully process the infected artefacts or keeping the malware infection *in situ* within a disk image instead of cleaning it. Both Ben Fino-Radin, associate media conservator at MoMA, and Don Mennerich, digital archivist at NYU Libraries, were sympathetic to the idea that malware infections may be of interest to future researchers. However, staff at other institutions may not have the same level of sensitivity or the same long-term outlook. Gruning's research raised the possibility that other institutions may believe that saving malware may be too troublesome and that removing it does not present a problem.[152]

---

[152] Gruning, "Rethinking Viruses in the Archives."

Current Archival Workflows

I will present a brief walkthrough of the workflows for imaging born-digital artefacts from the archivists and conservators I spoke with, and then discuss some of the specific issues raised by malware infections. In the case of imaging a hard drive, the drive is usually connected to a computer station designed for creating disk images (a common turn-key system is called F.R.E.D., Forensic Recovery of Evidence Device, used by both law enforcement agencies and archives, however institutions like NYU Libraries have also built their own imaging stations). Hard drives are attached to the imaging computer via a write blocker so that no data is written to the disk during the imaging step. This ensures that the disk image is a bit-for-bit copy of the data that exists on the hard drive being imaged. Typically, as soon as the artefact is imaged, the disk image is scanned for viruses and other malware. Currently both MoMA and NYU Libraries use forensic disk image formats such as the E01 format (also known as EWF [Expert Witness Format] or the EnCase format).[153]

NYU Libraries uses Forensic Toolkit Imager (FTK) to create its disk images. After imaging, FTK can scan for malware and flag all infected files. Mennerich said that FTK's flagging does sometimes result in false positives, so flagged files must always be examined more closely. Once the disk image is created, it is analyzed by the archivist or conservator, cataloged, and then packaged with its associated metadata into a Submission Information Package (SIP). The SIP is then sent to (or "ingested" into) the institution's digital repository for long-term storage.

---

[153] Fino-Radin, Ben. In conversation with the author. Phone call, February 1, 2016; Mennerich, Don. In conversation with the author, December 3, 2015.

Virus scanning of a SIP is often considered an integral part of digital preservation activities.[154] Digital preservation software (such as Archivematica and Preservica), which helps an institution prepare SIPs for ingest, typically offers virus scanning as a standard part of its feature set. Both NYU and MoMA use Archivematica to automate the ingest process. For institutions that use Archivematica, the software has a virus-scanning step (the default software it uses is ClamAV) which cannot be disabled by default. As of yet, Archivematica will not allow the ingest of SIPs determined to be infected with malware. The software will quarantine malware-infected SIPs. However, Justin Simpson, a software developer at Artefactual, the company that develops Archivematica, stated that, technically speaking, modifying the system to ignore the results of the virus scan and accept ingest of malware-infected files would not be overly complicated.[155]

Conservation Principles

The twin principles of the conservation of cultural heritage items—that "all alterations should be well documented and should be clearly distinguishable from the original object" and that "all interventions with the object should be fully reversible"[156]—ought to apply to malware-infected digital artefacts within collections. It should go without saying that any malware removal must be thoroughly documented, but in addition, creating an unaltered copy of the infected hard drive (as a disk image) makes malware removal theoretically "reversible" in

---

[154] See, for example, Megan Phillips et al., "The NDSA Levels of Digital Preservation: An Explanation and Uses" (Library of Congress, n.d.), http://www.digitalpreservation.gov/documents/NDSA_Levels_Archiving_2013.pdf.

[155] Justin Simpson, "Re: Research on Archivematica," March 28, 2016.

[156] "Conservation (cultural Heritage)," *Wikipedia*, accessed May 13, 2015, https://en.wikipedia.org/wiki/Conservation_(cultural_heritage).

that the institution could return to the infected copy to examine the state of the artefact before the malware removal or quarantine.

The actions taken by antivirus software are typically neither reversible nor documented using methods that are standardized or comprehensible to archivists or researchers. How antivirus software reacts when it detects malware depends on the malware encountered. Antivirus software typically deletes or modifies files (removing the malware code from an infected file) during virus removal. In other cases, the antivirus software will "quarantine" files. This means moving the files to a special folder that only the antivirus software has access to. In both cases, this modifies data on the hard drive, which may change the evidentiary value of the artefact.

Further research that must be undertaken into antivirus software could include what kinds of embedded metadata of files (such as created and modified dates) antivirus software alters when it quarantines or cleans files.

Authenticity and Trustworthiness for Born-Digital Artefacts

Does removing a malware infection compromise the authenticity of a born-digital artefact? In answering this question one must consider that contemporary scholarship that uses archival materials is "intimately bound up not only with the legitimacy of the source materials that formed the basis of the initial scholarly investigation but also with the reliability of the internal systems by which the repository documents how and when the items were acquired, their provenance, and the circumstances of their storage, and organization once on-site."[157] These

---

[157] Kirschenbaum et al., *Digital Forensics and Born-Digital Content in Cultural Heritage Collections*, 32–33.

internal systems of record keeping and documentation provide "evidence, as it were, about the documents in question."[158] This context makes the documentation of a malware infection more critical as it is part of the original state of the artefact when it arrived at the institution.

As good stewards, cultural heritage institutions must ensure that the artefact has not been altered in a way that compromises its authenticity or evidentiary value. Authenticity can be represented on a continuum for born-digital artefacts because of the ability, and often necessity, of altering them.[159] Greater potential also exists for altering digital artefacts as they pass through many hands—metadata which helps establish trustworthiness can be altered by "the very act of access."[160] When writing about authenticity for born-digital artefacts Clifford Lynch has argued that, "it is important to recognize that trust is not necessarily an absolute, but often a subjective probability that we assign case by case."[161]

Given that authenticity for born-digital artefacts exists on a continuum, is there a point at which a specific record starts to lose validity? Could the removal of malware constitute a change to an artefact that begins to erode its authenticity? Jane Gruning, who states that "reconstructed 'clean' files are not necessarily authentic," believes the removal of a malware infection compromises the item's historical value.[162]

When speaking about bit-for-bit accuracy, the answer appears clear—the removal or quarantine or malware is changing the bitstream. It is altering the data of the artefact. The fact that cultural heritage institutions would go to such lengths to preserve the artefact without

---

[158] Ibid.
[159] Julie McLeod and Catherine Hare, eds., *Managing Electronic Records* (London: Facet, 2005), 52.
[160] Kirschenbaum et al., *Digital Forensics and Born-Digital Content in Cultural Heritage Collections*, 27.
[161] Quoted in Kirschenbaum et al., *Digital Forensics and Born-Digital Content in Cultural Heritage Collections*, 27.
[162] Gruning, Jane. "Rethinking Viruses in the Archives."

alterations, such as using write-blockers and digital forensics software, means that this community has a strong desire to preserve the data on a born-digital artefact exactly as it was received. Perhaps the use of these methods also indicates that the cultural heritage community believes that any changes to the data may compromise the authenticity of the artefact. Given these assumptions, should the removal of malware constitute an exception to these generally agreed-upon principles? One could argue that malware removal technically does compromise an item's authenticity (in the sense that it alters the bits of a hard drive), but, in most cases, does not do so in a way that is significant. One could also argue that while malware removal may compromise an item's authenticity, removal is a necessary evil in order to make access to the artefact possible (a malware infection may make access to the contents of a hard drive or disk unsafe, difficult, or impossible).

Fino-Radin cautions against seeing technical purity as necessary for authenticity, especially when considering how to provide access. He gave the example of exhibiting Nam June Paik's work *Zen for TV* (a cathode ray television altered by the artist to display only a single line) without using the original cathode ray tube. The original tube cannot be expected to function forever and must eventually be replaced with something that has an equivalent behavior.

When accessing born-digital artefacts like hard drives, researchers typically look at a limited subset of the contents of the drive. In addition, different institutions will have different opinions on the level of technical purity required to maintain authenticity during access. Nevertheless, changes that were made to born-digital artefacts, especially those made in order to publicly present the artefact, should always be thoroughly documented, just as they were for the

Paik piece. Providing access to malware-infected digital artefacts will be discussed further in Chapter 7.

Malware in Archival Collections

While working jointly on processing the Susan Kare collection, MoMA and SFMOMA encountered malware.[163] The Kare collection consists of approximately 350 floppy disks and a boot-sector virus was discovered on several of them. When SFMOMA sent disk images to MoMA in New York City, the virus was detected immediately by MoMA's IT department. Because these disk images are part of MoMA's permanent collection, the IT department has not intervened to remove the malware, pending a decision from the conservation department.

At the time of my interview with Fino-Radin, the disk images from the Kare collection had not been ingested into MoMA's digital repository for reasons unrelated to the existence of malware. If the virus-infected files could be pushed past Archivematica's ingest workflow into MoMA's digital repository, Fino-Radin does not see a huge security concern in the case of this infection. The malware is a boot-sector virus specific to floppy disks. In addition, due to the nature of MoMA's repository architecture, there is little risk of cross-contamination in the repository. Storage is LTO-based and write once only—once a file is written to a tape, it cannot be overwritten. In addition, only very specific automated computer systems can write to the tapes and any data that is written to the tape has gone through a virus scan. However, this system presents a problem for the infected disk images from the Kare collection. If these images cannot be ingested into the repository, there are limited options about what can be done with them.

---

[163] Susan Kare is an artist and graphic designer who created many of the early Apple Macintosh icons.

At NYU, Mennerich encountered malware within a collection on a Macbook Air. The

Macbook's hard drive contained sixteen .EXE (Windows executable) files identified as malware.

Since the laptop was a Macintosh machine, it was not actually infected. When the drive was

imaged, the files were flagged through FTK software. Mennerich researched the viruses and

discovered that they were relatively old, but without more information he still does not know

how they were copied to the hard drive in the first place. Mennerich also pointed to the fact that

analyzing a malware infection can become a time-consuming process. NYU Libraries has kept a

bit-for-bit copy of the infected hard drive for preservation purposes, but may not make it fully

accessible.

Both NYU's and MoMA's situations underscore the point that no standard exists for

handling malware-infected artefacts, documenting the malware found on items within

collections, and whether this information should appear in the accession record for the item.

While working as an archivist at Johns Hopkins University, Christie Peterson

encountered boot-sector viruses:

> One of my accomplishments was to image all of the removable media in the archives and
> manuscripts collections, and included in that were several examples of disks from the
> 1990s that contained boot sector viruses. This was discovered during the accessioning
> workflow, during which I ran malware scans on the disk images. However, I did not
> undertake any steps to clean the disks or the disk images. The primary reason for this was
> that the infected disk images can be stored safely, and I was using imaging primarily as a
> short-term preservation tool; that is, my concern was primarily with getting the content
> off of the degrading media. Additionally, I found that the contents of the disk images
> could be safely ingested into FTK, which was what we were using to process the
> collections, without triggering the viruses.[164]

But Peterson's work with imaging infected disks was not without peril:

> I also encountered boot sector viruses on at least one 1990s era floppy that I attempted to
> image while I was a project archivist in the university archives at Princeton...somehow

---

[164] Christie Peterson, "Re: Malware Preservation," March 14, 2016.

the malware scan triggered the virus and it completely bricked [rendered unusable] the Windows machine I was working on—the machine was old enough that IT chose to just replace it rather than reimage it. The actual disk from that situation was deaccessioned.[165]

Other institutions have preserved infected digital artefacts as well: "Stanford preserves the whole image, virus and all...Emory [University]...also preserves the disk image with viruses, but excludes viruses from any exported files [for access]."[166] Thus far, the staff I spoke with have not had to tangle with contemporary malware, which could potentially spread over a network or create a botnet.

Fino-Radin and Mennerich do not seem overly concerned about the malware they discovered given that it was spread through floppy disks and does not make network connections. This perspective makes sense as any malware their institutions store will be wrapped in a disk image and written to an LTO tape which will be placed in deep storage. Malware within a disk image that is written to a data tape poses no risk until the tape and image are accessed again. I will discuss risk assessment for malware access in more detail in Chapter 8.

Best Practices for Processing Malware-infected Materials in Archival Collections

The purpose of discussing best practices is, in part, to identify the ideal procedures institutions should aspire to, as well as the risks of not adhering to these practices. Given the enormous amount of economic resources, time, and staff involved in digital preservation, institutions with differing priorities and levels of resources may need to make compromises and deviate from best-practices.

---

[165] Ibid.

[166] Julia Kim, "Capturing a Shadow: Digital Forensics Applications with Born-Digital Legacy Material," *NDSR-NY*, October 17, 2014, http://ndsr.nycdigital.org/capturing-a-shadow-digital-forensics-applications-with-born-digital-legacy-material/.

For malware-infected digital artefacts, saving two versions of a disk image or file—the "clean" version and the "infected" version—appears prudent as a best practice. This method would preserve the artefact in its original form and allow institutions to defer the decision about what to keep until later, while the physical media is still readable. For example, if the archive decided to only keep a "clean" disk image of a hard drive, but then changed its mind ten years later, the original drive may no longer be readable.

Keeping the infected disk image and documentation about the infection ensures that researchers who are interested in studying the malware infection have the recourse to do so. Saving two disk images requires double the storage space for each infected item, but this may not amount to a large amount of space overall as malware infections within archival collections remain relatively rare. Saving a disk image of just the infected version may be acceptable as well, as many institutions I discussed have already done. As a best practice, disk images should be created with forensic formats like E01. These formats offer many advantages for storage, documentation, and access, which will be discussed in more detail in Chapter 7.

Another important point to consider is how many files are actually infected. If a Word macro virus, for example, is just affecting one file, does the institution still need to save two complete disk images? How can one easily determine whether the macro virus has affected other parts of the hard drive? Is there a threshold at which the institution should keep the entire disk image? When discussing whether or not to store a complete disk image, Gareth Knight suggests that some archives adopt "a middle-ground approach" where they store complete disk images for certain collections that are especially important (or where disk images are small), while for other

collections or artefacts they only use selected data extracted from the artefact for the SIP.[167]

However, this "middle-ground" approach is not entirely accepted within the cultural heritage

community, as many within the community would argue that creating disk images for all

born-digital artefacts entering collections is the best practice.

Once an institution ingests a file into its digital repository, such as a disk image or a web

archive, the ingested file should not change after the fact. It should receive regular inspections of

its checksum to ensure that no bits have changed. The routine checksum scans ensure the

integrity of the files. However, conducting antivirus scans after ingest and removing malware

has the potential to compromise the integrity of the files and break the unaltered chain of

custody.

---

[167] Gareth Knight, "The Forensic Curator: Digital Forensics as a Solution to Addressing the Curatorial Challenges Posed by Personal Digital Archives," *International Journal of Digital Curation* 7, no. 2 (December 6, 2012): 40–63, doi:10.2218/ijdc.v7i2.228, 50.

Chapter 5: Malware Preservation Strategies and Challenges

On the one hand, the care in curating conceptually tends toward the presentation of the static: collecting, archiving, cataloging, and preserving in a context that is both institutional and architectural. There is a stillness to this...the care of stillness, within walls, behind glass, is a historical stillness...But there is always an excess in curating, an opening, however wide or narrow, through which the unexpected happens. —Alexander Galloway and Eugene Thacker, *The Exploit*

This chapter discusses malware preservation strategies and where they diverge from strategies for the preservation of commercial or mainstream software. Many of the core principles of software preservation apply to malware and any institution engaged in a malware preservation project should be in conversation with the software preservation community. However, this section primarily focuses on the differences between malware preservation and the preservation of other types of software.

Malware offers unique challenges for preservation above and beyond just saving code and ensuring that the program being preserved can function in the future. Malware preservation will require the execution of multiple strategies simultaneously as individuals do not solely experience malware as a line of code or as an infected disk. Persistence for malware means preserving as much of the experience or event as possible.

The majority of this chapter concerns suggestions for institutions interested in intentionally collecting materials related to a particular virus or to the history of malware. These suggestions go beyond materials that are normally deposited in an archive, museum, or library and would require staff members to proactively seek out malware samples, related files, and other ancillary materials such as virus creation tutorials and media reports on malware infections.

As with any other collection, preservation and access strategies for malware collections will always depend on the actual and projected needs of the researchers. Do researchers want to inspect the malware's code? Do they want to see the malware demonstrated? Are they hoping to view a malware infection "in the wild"?[168] Malware in a collection may have different forms of provenance and thus a different kind of research value. Malware obtained "in the wild" (for example, through the donation of an infected computer that the owner did not even know was infected) has its own value as evidence versus malware that was obtained directly from a creator. In effect, "the cultural, historical, scientific, and economic reasons for acquiring and retaining born-digital materials—along with the intended or project use cases—have a significant effect on both archival planning decisions and development of the technical infrastructure necessary to support retention and access over time."[169]

Some researchers, particularly in the humanities, may not be as interested in browsing the code as in having a general idea of how the malware worked, what it looked like to computer users, how it impacted an individual user, and the societal reaction (or indifference) to its release. Others may want to witness the malware demonstrated on an actual computer or may want to interact with the program. Both styles of usage require distinct preservation and access strategies.

Malware Release as an Event

Few releases of software have generated the impact (on both individual users and the media) of the most prominent malware epidemics. When Adobe releases the latest version of

---

[168] "In the wild" refers to an infection that occurred through a person's everyday use of their computer. This is in contrast to malware that was intentionally placed on a computer for testing or analysis.

[169] Woods, Lee, and Garfinkel, "Extending Digital Repository Architectures to Support Disk Image Preservation and Access," 61.

Photoshop, technology experts and journalists generally do not panic and make doomsaying

pronouncements about the potential ramifications of the public release of the software. The

release of certain virulent malware, on the other hand, may take on the character of a landmark

event and receive widespread news coverage.

Jussi Parikka suggests viewing malware infections as events "that are overflowing in

their rigid territorializations (as malicious software)."[170] In other words, the experience of the

malware's release transcends the payload of the malware, whether the virus deletes data or steals

passwords. The media reaction to the ILOVEYOU virus transformed the infection of a large

number of individual computers into a hyper-mediatized spectacle and the virus is remembered

as one of the largest international computer pandemics. Biennale.py's authors consciously

designed its release as an event (although many other malware creators probably hope that the

release of their own virus or worm will become spectacular as well). The reaction to

Biennale.py's release may have said more about the media's ability to whip up a frenzy than

about the virus itself, whose payload was negligible. The simultaneous release of the code on

t-shirts and posters performed the act of "going viral" outside of the territory of computers and

the internet.

To draw an analogy, researchers studying the history of the 1902 smallpox epidemic

would not simply be content to look at the genetic code of the virus, but would want a fuller

picture of the place where the epidemic occurred (Boston, Massachusetts), the societal factors

that caused it to spread, and the methods of those who worked to fight the disease.

---

[170] Parikka, *Digital Contagions*, 5.

Unfortunately, events or "lived experience," such as a baseball game, a performance art piece, or the production of a movie are notoriously hard to preserve and archive.[171] In these cases, as with malware infections, it is often unclear what exactly must be saved. Furthermore, in certain respects, the archive will always fall short of preserving these events as it can only capture their tangible residues (such as the scorecard of a baseball game or a news report of a new computer virus outbreak).

A Time-based Media Art Approach

The concerns facing an institution intentionally acquiring a virus or worm into its collection may be similar to those of a museum acquiring a work of "electronic art," now more commonly referred to as time-based media art: "electronic works are usually difficult to capture, and...in many cases it's not even clear what elements *need* to be captured."[172] Answers to preservation and conservation questions may need to be determined on a case-by-case basis. Many context-specific decisions and risk assessments may need to occur. Approaching the preservation of malware with a similar mindset to preserving time-based media expands one's thinking outside of the question "What is the object to preserve?"

Storing a physical object like a disk will prove insufficient for the long-term preservation of both malware and time-based media. Howard Besser instead suggests a more practical approach to preserving time-based media including, "trying to ascertain what the work really is,

---

[171] This was suggested by JP Dyson's presentation "So How DO You Preserve a Video Game?" (Pressing Restart: Community Discussions on Video Game Preservation, NYU Game Center, September 28, 2013).
[172] Howard Besser, "Longevity of Electronic Art," accessed May 7, 2015, http://besser.tsoa.nyu.edu/howard/Papers/elect-art-longevity.html.

trying to make the critical portions of it persist over time, and saving ancillary materials that become critical to understanding that work."[173]

Similarly, in the case of malware, determining the critical components to preserve it and the ancillary materials necessary to understand it are important because the long-term preservation of computer hardware will prove completely impractical. Computer hardware as well as magnetic and optical media (like hard drives, disks, videotapes, and DVDs) face the twin dangers of both physical deterioration and obsolescence. Most spinning-disk hard drives were not designed to last more than a decade or so. Therefore, the periodic need for file refreshing and migration to avoid obsolescence and inaccessibility apply equally to malware, software, and time-based media art.[174]

Like some time-based media art, a piece of malware may have very specific hardware and software dependencies. While commercial software may strive for compatibility between different operating systems, a piece of malware may have a very low degree of compatibility and may only run in one particular version of the operating system. Even within the narrow category of viruses and worms, different examples have different methods of infection, which means they exploit different aspects of a computer system. Individual pieces of malware may be unique in their behavior or their method of infection—only a single piece of malware may have ever been written to exploit a particular security vulnerability in an operating system or web browser. This uniqueness complicates risk analysis and preservation workflows—much like conservators conduct a thorough analysis of time-based media works to determine deliverables from an artist

---

[173] Ibid.
[174] Ibid.

or gallery, an institution may need to conduct an analysis on each piece of malware it decides to collect.

A preservation plan and risk assessment for each individual piece of malware will be time-consuming, which means that institutions purposefully collecting malware would be wise to carefully select which examples they are going to save. However, many of the more "benign" viruses and worms from the 1980s and the 90s, some of which do not have a payload, may require a less thorough risk assessment as their effects may be well-documented and most do not make network connections. For institutions beginning to build malware collections, it may be wise to start by collecting these viruses and worms.

Saving Code

The most obvious strategy for preserving malware is saving code (ideally source code), or other files packaged with the malware.[175] David M. Berry describes computer code as "a literature, a mechanism, a spatial form...a repository of social norms, values, patterns, and processes."[176] As a "repository of social norms, values, and patterns" code can be of significant interest to researchers in the humanities.

At a basic level, code is a series of logical instructions to a computer that govern how a program runs. Examining code can reveal behaviors that are unseen or unknown to an end-user viewing the output of the executed code. By analyzing the code, one can understand time delays,
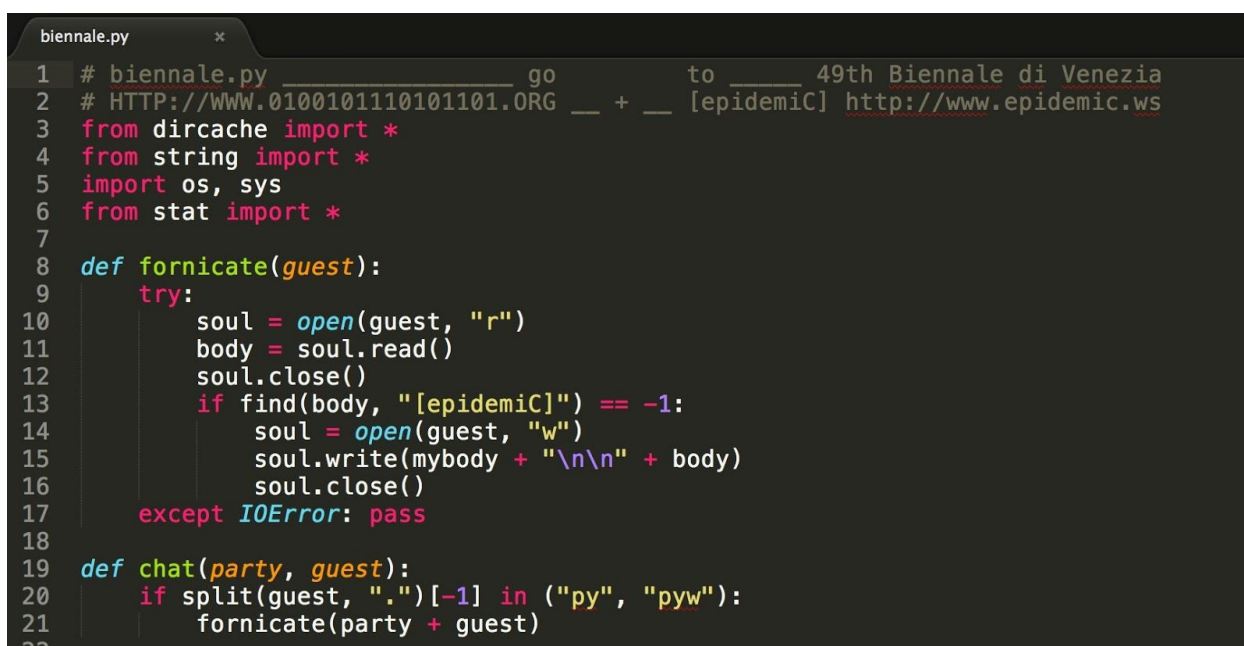
---

[175] Source code is the original human-readable code written by the programmer before it is compiled. Most source code needs to be compiled so that a specific CPU (for example, an x86 processor or a PowerPC) can run it. Source code often contains comments from the author and other natural language.

[176] David M. Berry, *The Philosophy of Software: Code and Mediation in the Digital Age* (Basingstoke, Hampshire ; New York: Palgrave Macmillan, 2011), 65.

restrictions on how the malware runs, network connections that are made, or payloads that may

not be immediately clear when running the code.

One method through which programmers can express opinions or document logic flow

with code is through commenting. Comments are text within code written in natural languages

(like English). Comments, which generally only appear within source code or scripts, are not

designed to be executed by a computer, but can be read by people. Sometimes a programmer will

use comments to explain their code to other programmers, or leave notes for themselves. In

Figure 5.1 below, comments appear in lines of code that begin with a hash (#) symbol.

```
biennale.py                ×

1   # biennale.py _____ go        to _____ 49th Biennale di Venezia
2   # HTTP://WWW.0100101110101101.ORG __ + __ [epidemiC] http://www.epidemic.ws
3   from dircache import *
4   from string import *
5   import os, sys
6   from stat import *
7
8   def fornicate(guest):
9       try:
10          soul = open(guest, "r")
11          body = soul.read()
12          soul.close()
13          if find(body, "[epidemiC]") == -1:
14              soul = open(guest, "w")
15              soul.write(mybody + "\n\n" + body)
16              soul.close()
17      except IOError: pass
18
19  def chat(party, guest):
20      if split(guest, ".")[-1] in ("py", "pyw"):
21          fornicate(party + guest)
22
```

Figure 5.1: Biennale.py's source code viewed in a text editor. Comments on lines 1 and 2 identify the creators. Comically named functions (such as "fornicate") and variables (such as "soul") are visible as well. (0100101110101101.org)

Other forms of communication within code include the naming of variables, objects, and

functions. Naming these attributes is often not arbitrary. Programmers create names that

represent the variable or function's purpose within the code or occasionally they use these names

for comic or satirical purposes. It remains to be seen if there are other aspects of the way code is arranged or written that reveal norms, values, and patterns of the creator.

Code may be used in the future to help decipher who programmed the malware. Discriminating investigators in the field of software forensics can parse the code for evidence of its author: "quirks and tics in a suspect program's language (the source code of a virus, say) are exploited to trace them to an individual human author, much like forensic linguists exploit stylistic features to attribute anonymous texts."[177]

Aside from examining source code, one can also examine binary or compiled code. This typically occurs if source code is unavailable. One method of viewing binary code is through a hex dump. A hex dump displays the contents of a file in hexadecimal code (also called hex), a way of representing binary code that is easier for humans to read. Most hex viewers can translate any ASCII (a way of representing written text by a computer) found within the hex. Analyzing the hex can reveal messages within the code. For example, within the Blaster Worm's (2003) code is a message for former Microsoft CEO Bill Gates.

---

[177] Kirschenbaum, Matthew. "Hello Worlds." *The Chronicle of Higher Education*, January 23, 2009. http://www.chronicle.com/article/Hello-Worlds/5476.

Figure 5.2: A hex dump of the Blaster Worm. Translation of the ASCII code reveals a message from the worm's creator. (Wikipedia)

The hex code of the Brain Virus revealed the names of its creators as well as their company's address and phone number.

Computer security analysts will sometimes attempt to decompile binary code to make it human-readable as source code. While decompilation can provide security analysts with readable code, sometimes the decompiler cannot properly reconstruct source code from the binary and may produce indecipherable characters or lines. In addition, any comments the programmer wrote in the original source code will not appear in the decompilation as comments in the code are not compiled.

Malware does not always exist as a compiled executable file, it could be written in an interpreted language (such as Biennale.py, which was written in Python) or as a script. Word macro viruses or email worms that exploit Microsoft Outlook were typically written as Visual Basic scripts. Thus, anyone who understands the scripting language could then study the malware's code without needing to decompile it.

Through examining code, one can perhaps understand how malware coders learned how to create viruses, or what inspired them to do so. In the same way that one can examine numerous HTML documents and learn that they originated from a Dreamweaver template (they usually contain the name of "Scaal" a fake coffeeshop used in Dreamweaver tutorials),[178] certain signature code, or text could point to previous pieces of malware from which the malware in question arose. Malware can also be intertextual; new malware could be an homage to something older. Tracing the history of malware vectors (like disks, emails, and web pages) provides historical perspective into how users were sharing files—common storage and transmission technologies that malware creators were prepared to exploit.

While examining the code of viruses and worms can be helpful in understanding the program or its origin in certain circumstances, at other times malware code may be extremely misleading. For example, the code in Dark Avenger's V2000 and V2100 viruses read "Copyright Vesselin Bontchev." Bontchev is a well-known computer security researcher and the text that the viruses' author inserted into the code may have been meant to damage Bontchev's reputation. Information contained within code must always be cross-checked with information from other sources. If code is all that a researcher has for reference, any inferences that they make may be questionable.

It is also important to note that "virus collection samples exist in many forms. The original source code, the assembled object file, the linked first generation binary, an infected victim binary, an infected goat file,[179] debug script, UUE encoded script, boot sector images, disk

---

[178] Kirschenbaum, *Mechanisms*, 121.

[179] Goat files are used to analyze the code of a virus. "These files have an exactly known layout and length and when infected by a particular virus the difference between the clean and infected goat file will tell researchers a lot about the virus." See Cicatrix, "Collecting Computer Viruses: Fun or Folly?"

images...incorrectly compiled code and disassembled source code."[180] These different kinds of files all have different purposes and levels of utility in malware analysis. An archive intentionally collecting malware should strive to create documentation and workflows that account for these differences.

Preserving malware code is absolutely vital, but saving code is only one tool in a preservation toolkit. Computer or technology museums are likely to have more impetus to save code, particularly source code. However, code in isolation may prove difficult to use for research several decades in the future when the expertise may not exist to fully interpret it. Few experts on interpreting older languages like COBOL and ALGOL exist today. Interpretation can be especially difficult if the researcher is not even sure what the software or script is designed to do, which will likely be more common for malware than for conventional software, given the anonymity of many of its authors and their reluctance to share information about its capabilities. When asked about saving code, computer programmer Steve Lamb made the observation that, "almost more important than saving the code might be saving a description of what the code does."[181]

Code that cannot actually be run on a computer has limited utility. Simson Garfinkel, Chief of the Center for Disclosure Research at the US Census Bureau and an authority on digital forensics and information security, pointed to the importance of the capacity to "'test' or demonstrate the malware in a controlled environment," which relates to his concern that malware code may not be worth preserving if a runtime environment cannot be preserved as well.[182] As computer hardware and operating systems are constantly changing and becoming obsolete, the

---

[180] Cicatrix, "Collecting Computer Viruses: Fun or Folly?"
[181] Steve Lamb, In conversation with the author, April 12, 2015.
[182] Simson Garfinkel, "Re: Research on Malware," March 8, 2015.

ability to demonstrate historical malware will almost certainly depend on hardware or software emulation or virtualization to recreate a historical computing environment.

As a benchmark for determining which elements related to malware should be retained, I examined the 2016 "Recommended Formats Statement" released by the Library of Congress for copyright submission. This statement included recommendations for software and electronic gaming. As a best practice, the Library requested documentation, source code, operating system, and platform (if the software was for a standalone item like a video game console). The Library also advises that

> Metadata that specifies which compiler was used to create the final code for commercial release—including the version number and build number of the compiler software—must be included. If the compiler is unique to the project or company...then a copy of compiler software in the specific version and build used to create this version of the software, along with specifications of the platform the compiler ran on, must be included in the submission.[183]

Obtaining information about the compiler used or obtaining the source code may be impossible for malware. A program's source code is typically obtained from the original programmer, and malware coders, who often wish to remain anonymous and do not want to take responsibility for any particular piece of malware, will likely not be willing to share their source code.

Non-Linear Interactivity

Some pieces of malware are interactive and resemble video games (such as the Casino [1991] and the Monte Carlo DOS viruses, which challenge the user to a card game). The Happy Birthday Joshi Virus (1990) activates on the birthday of the virus's creator and asks the user to type in a birthday greeting. These viruses are nonlinear in their operation and need to be truly

---

[183] "Recommended Formats Statement – Software and Electronic Gaming and Learning," *Library of Congress*, accessed February 14, 2016, http://www.loc.gov/preservation/resources/rfs/softgame.html.

interacted with in order to be fully understood. Interactive malware can have a hidden payload

that will only be revealed when it is played. This is unlike software like Microsoft Word, which

is relatively static in its presentation to the user.[184] Video walkthroughs of interactive viruses and

worms can help a viewer understand the content that is not readily apparent. The majority of

malware has little interactivity and when it does, the interactivity is never as immersive or

multifaceted as that of most video games.

Figure 5.3: A screenshot of the payload screen of the Casino Virus. The virus challenges the user to a card game where they can place bets in an attempt to win back the data on their hard drive. (Malware Wiki)

Significant Properties

Significant properties are "those properties of digital objects that affect their quality,

usability, rendering, and behaviour."[185] Significant properties are especially important to keep in

---

[184] Becks Hernandez-Gerber has made this comparison between software like Microsoft Word and video games.
[185] Margaret L. Hedstrom et al., "'The Old Version Flickers More': Digital Preservation from the User's Perspective," *The American Archivist* 69 (Spring/Summer 2006): 159–87.

mind when emulating or migrating software or files. Just like two different art installations or

interactive CD-ROMs, two pieces of malware may not have the same significant properties. For

some malware, a significant property might be the display of a payload screen (such as the

Phantom 1 Virus [1994]). For other malware, significant properties may include some degree of

interactivity (Monte Carlo and Casino viruses). Significant properties could also include the

sound effects or the music that the virus plays. For malware that tries to hide itself from the user,

the significant properties may include how the malware affects files on an infected computer, the

kinds of network connections the malware makes, or how it reacts to antivirus software. Some

significant properties may relate to the events and materials surrounding the malware's release

(Biennale.py). Archivists, librarians, and conservators may have to carefully analyze each piece

of malware to determine its significant properties.

Saving Ancillary Materials

In preserving malware there are many kinds of ancillary materials to consider. Saving

media reports (such as online and offline magazine and newspaper articles, blog posts, and the

like) on the release of malware will provide additional context for the release as an event.

However, given the diverse motivations and the secretive culture of malware coders, these media

sources are prone to reporting inaccuracies and exaggerations. Douglas Thomas gives the

example of the hyperbolic reaction to the 1995 release of SATAN (Security Administrator Tool

for Analyzing Networks) in the mainstream media, which compared the program to a rocket

launcher or an automatic rifle. Unfortunately,

> public reaction to hackers both tells us a great deal about the public that is reacting and,
> ironically, shields us from an understanding of the complexities and subtleties of the
> culture of the computer underground. By simply equating hackers with the tools they use,

the media and popular representations of hackers have failed to understand or account for even the most basic motivations that pervade hacker culture.[186]

Nevertheless, saving ancillary material and intellectually connecting them to assets like malware code or infected disk images may be critically important for the preservation of malware. Computer security forums, newsgroups, mailing lists, Computer Emergency Response Team (CERT) advisories, and information from other computer security organizations and companies can assist with the identification of malware and contain important information about how the viruses work and how they can be removed. Ancillary documents can also help an institution understand how to catalog their collection.

As mentioned previously, the release of malware often invites great controversy within both the computer mainstream and underground. The reactions to the releases of the Morris Worm, Back Orifice, and SATAN were especially intense. Saving ancillary material helps researchers trace the various discourses and narratives that were in conflict during the program's initial release. Tracing these lines of argument can have much greater implications for how the history of software and the internet is written.

Malware creators may post communiques or comments about their malware online. Websites, social media networks, and blogs will often quickly respond with posts, articles, and photos related to a new malware infection, which makes some form of web archiving especially relevant for preserving the history of malware, or finding contextual material for a particular piece of malware. There are several websites, such as VX Heaven, which contain extensive information about virus programming as well as virus samples.

---

[186] Thomas, *Hacker Culture*, 9.

Institutions must keep in mind that communiques from creators or emails written by those infected, may be extremely ephemeral (for example, a video message posted by a malware developer may be quickly taken down from YouTube).

Keeping logs of infected computers and networks may provide important information, even if the infected machines cannot be saved long-term. It may also be possible to freeze command and control servers of botnets and image them for preservation.[187] Computer security analysts use saved files throughout a computer system (not just the malicious code itself) to understand a malware attack or identify an intruder. Therefore, "much hacker and cracker lore is given over to the problem of covering one's 'footsteps' when operating on a system uninvited; conversely, computer security often involves uncovering traces of suspicious activity inadvertently left behind in logs and system records."[188] This makes saving full disk images and server images important for future malware research, especially where no one has gone through a detailed forensic analysis of the threat. In general, detailed analysis is only conducted on malware during incident response (and only to the most high-profile or most threatening malware). Much malware exists that has received little to no attention or professional analysis.

When considering ancillary documents and artefacts to save, institutions should also consider saving programs like virus creation kits, droppers, and polymorphic engines.[189] Virus creation kits usually have a graphical user interface and allow a user to create their own virus based on some set of predefined attributes and behaviors. The program assembles the virus without the user having to write a line of code. Droppers are programs that install malware on a

---

[187] Dave Riordan raised this possibility in conversation with the author.
[188] Kirschenbaum, *Mechanisms*, 49.
[189] Polymorphic engines transform the code of a program while maintaining the exact same functionality. Viruses make use polymorphic engines to avoid detection by antivirus software.

target computer. Using droppers may be a way to avoid detection by antivirus software or install multiple pieces of malware on a single computer. While not viruses or worms themselves, these programs are certainly part of malware's ecology and have helped facilitate the spread of viruses.

An archive may also want to collect tools used on the other side of the struggle, such as various editions of antivirus software, virus identification tools (such as the one developed by Ivan Triffonov), virus analysis tools (like disassemblers, decompilers, and debuggers), virus simulators, as well as selections from the enormous corpus of scholarship related to understanding malware and protecting against it.[190]

As Cicatrix states: "Most virus creators are only too happy to share their knowledge with the rest of the world and will write extensive tutorials on a wide-ranging number of pro virus [sic] subjects. A lot of virus authors are self-taught and use many of these tutorials as learning material."[191] Virus tutorials are another fount of historical information for understanding both the viruses themselves and the community around them.

Using Multiple Strategies Simultaneously

In addition to using multiple preservation strategies within an institution, one can also conceptualize two simultaneous mindsets for the collection and preservation of malware. One strategy would involve comprehensively curating and preserving selected pieces of malware (for example, if the Computer History Museum decided to preserve the WANK Worm), where the institution commits to preserving a runtime environment, associated materials, and perhaps infected hard drives and logs. The second, complementary strategy would revolve around

---

[190] See Bontchev, "Analysis and Maintenance of a Clean Virus Library."
[191] Cicatrix, "Collecting Computer Viruses: Fun or Folly?"

larger-scale sweeps, for example, Jason Scott's uploads to the Internet Archive and to his own

website, textfiles.com. Textfiles.com alone contains 295 files with information about infections,

tutorials for programming viruses, bibliographies related to viruses, and examples of virus code.

For institutions interested in building large collections of malware, an additional

acquisition strategy involves examining large web archives, for example the corpus of the

Internet Archive, to locate malware and ancillary materials. Unfortunately, the Internet Archive

is too large to scan completely for malware and related documentation.[192] A strategy for scanning

selected areas of the archive that are likely to contain these materials could be developed.

Forging relationships with "white hat" hackers who might be saving malware may also

improve one's chances for collection. Former hackers who now work at computer security firms

may also be willing to donate or share code or files.

Additional Technical Challenges for Intentional Collection

Malware's code, especially among polymorphic viruses, is almost never static. For

example, the WANK Worm's code base changed as it infected more computers. For an

institution collecting this worm, the changes make it hard to determine which code to save.

Archivists may need to use a "snapshot" methodology of the malware in a number of discrete

stages, which could be similar to the archiving of a news website or blog; however, in web

archiving one is capturing new information being added to the site mostly by human beings,

whereas with a worm, the updates and changes often occur without human intervention or on a

set schedule. Comparing the changes between snapshots may be important for learning about the

---

[192] Jefferson Bailey, "Re: Viruses and Malware," May 3, 2015.

architecture of the malware and its behavior. Polymorphic viruses may require a number of

snapshots. Archivists interested in determining which snapshots are the most significant would

benefit from consulting with experts in malware analysis.

Authors of malware typically wish to remain unidentified (and many remain unknown),

which means that obtaining source code or reliable information related to the piece of malware's

development will often be impossible. What's more, if there is no definitively known creator,

ensuring the provenance, or a chain of custody for malware code or other related files may be

difficult. A well-documented chain of custody between a creator or a trusted party and an archive

may not exist.

Kirschenbaum et al. elucidate the challenges of online anonymity for cultural heritage

institutions: "If a writer dies without leaving any documentary record of her aliases, it might be

difficult for the repository to create a full picture of her online communities."[193] A malware

creator may go by several aliases and it may be impossible to determine their footprint in the

malware coding community or on the internet. Archivists, librarians, conservators, and

researchers must consider privacy issues as well; even if an archivist or researcher could

determine the true identity of a malware programmer, would it be ethical for the institution or

researcher to release this information publically?

Issues with Migration or Normalization of Files

Any institution intentionally collecting malware or in possession of malware-infected

artefacts must carefully consider how migration or normalization may affect the

---

[193] Kirschenbaum et al., *Digital Forensics and Born-Digital Content in Cultural Heritage Collections*, 55.

malware-infected files within their collections. Many archives will need to migrate or normalize

their files. Migration or normalizing means taking a digital file and transforming it into a

different format (for example, taking a Word document and turning it into a PDF). Migration

may occur in order to transfer the file to a more preservation-friendly format or to make the file

easier to access. However, migration does not guarantee the preservation of many characteristics

of the file, including a malware infection. Hedstrom et al., who studied file migration techniques,

discovered that:

> the original bitstream could be altered by bugs in conversion software, mishandling or
> failure of storage media, incompatibilities between the original and migrated formats, and
> changes in compression, file sizes, media density, and file names. Such changes may
> introduce errors...affect linkages to other files, such as metadata files, database
> directories, scripts, and URLs; or eliminate unique features of the original format that are
> not supported by the migrated format.[194]

In addition to the risks mentioned above, a "unique feature" lost in migration could be the file's

susceptibility to malware infection. Thus, a malware infection could be lost through a migration

process. If a Word document is infected with a macro virus (essentially a Visual Basic script

within the file), the infection would disappear if the document were converted into PDF because

the script would not be transferred.

Simson Garfinkel identified a unique technical challenge for making malware persist in

an archive, namely "the ongoing challenge of preventing it from being cleaned by antivirus

systems."[195] Collections containing malware could be especially vulnerable if the institution's IT

department does not consult closely with archivists, and inadvertently alters collection items by

removing or quarantining malware or malware-infected disk images.

---

[194] Hedstrom et al., "The Old Version Flickers More," 164.
[195] Garfinkel, "Re: Research on Malware."

Payload Screens and Video Capture

Payload screens can often speak volumes about the purpose of the malware, identify the author or authors, help date the malware, or provide other historical or contextual information. On October 17, 1989, staff at NASA logged onto their computers and were faced with this screen:

Figure 5.4: Payload screen of the WANK Worm. (NetSentinel)

Through the payload screen and the name of the WANK worm, one can surmise a particular political position. Researchers used information from the payload screen, among other evidence, to trace the worm's origin to Melbourne, Australia. "You talk of times of peace for all, and then prepare for war" is a lyric from the Australian band Midnight Oil, and "wank" is a slang term for

masturbation common in the U.K., Ireland, Australia, and New Zealand. Nevertheless, payload

screens are not transparent windows into the mechanisms or intent of the malware. For example,

WANK's payload screen claimed it was deleting files, yet no files were being deleted on

NASA's computers at all.

The original payload screen of the Spanska Virus read "Remember those who died for

Madrid No Pasaran! Virus (c) Spanska 1996." The text referenced a famous speech by Dolores

Ibárruti, a Republican heroine in the Spanish Civil War. A later variant of the virus displayed a

CGI animation of the surface of Mars along with the message "coding a virus can be creative."[196]

Using the virus's animation and the text, Spanska sent a message about the entire enterprise of

malware coding.

Figure 5.5: The payload screen of the Spanska.1500 Virus variant (also known as Mars Land). (F-Secure Labs)

---

[196] Hypponen, Mikko, and Peter Szor. "Spanska Threat Description." F-Secure, 1997.
https://www.f-secure.com/v-descs/spanska.shtml.

On the YouTube channel "danooct1" virus collector Daniel White infects his vintage Packard-Bell computer with Windows and MS-DOS viruses and takes video captures of the screen as he talks about the history of the virus. White then executes the virus until it unleashes its payload and describes how it is affecting his computer.[197] White gets his information from security websites such as F-Secure and Kaspersky (although he often consults older versions of the sites saved on Archive.org). For research purposes, he has also purchased old antivirus software off of Ebay. The old discs sometimes include virus descriptions, which he finds useful. He recently purchased antivirus software that came packaged with "a 400+ page book of nothing but DOS virus descriptions."[198]

---

[197] See https://www.youtube.com/user/danooct1
[198] Daniel White, "Re: Malware Preservation Research," April 4, 2016.
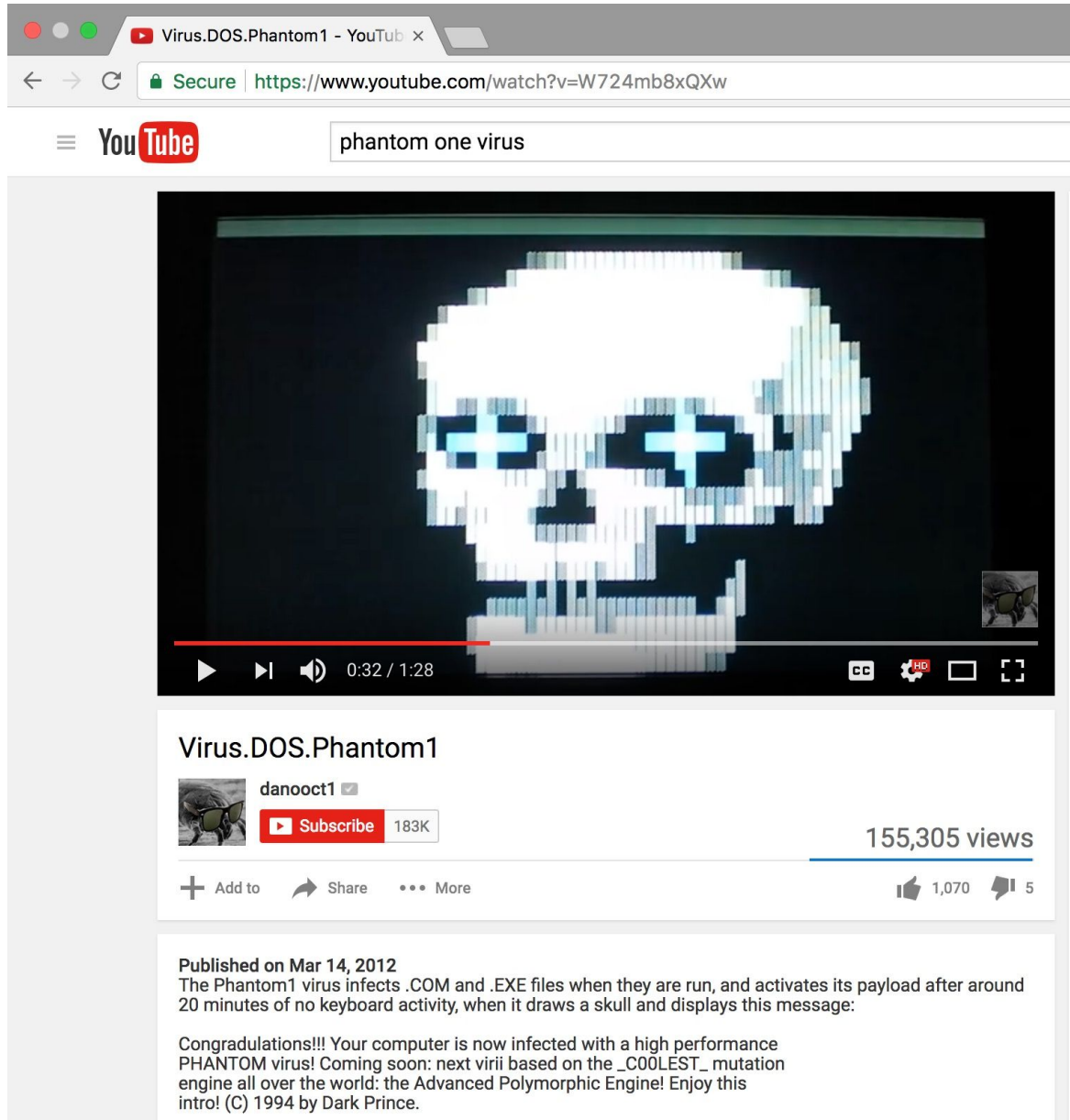
Figure 5.6: Daniel White's video of the Phantom 1 virus. The page includes a description of the virus's effects and the payload screen. (YouTube)

White's work has similarities to the Strong Museum's International Center for the

History of Electronic Games's (ICHEG) initiative to record the gameplay of console video

games and have the player talk about the game mechanics and controls as they are playing.

ICHEG considers this strategy as a "Plan B" if it cannot make the games persist through

emulation as there will be some record of the work.

In a video capture, the user can correct for misinformation displayed on a payload screen.

They could, for example, mention that no files are actually being deleted by the WANK Worm.

Digital video files have far fewer hardware and software dependencies and will likely be far less

complicated to preserve than executable malware. Video recordings of malware also do not

present any of the security concerns during access that live malware would. Similar to ICHEG's

philosophy regarding the preservation of video games, I believe that video captures represent a

viable Plan B for the preservation of malware.

Preserving the Process of Creation

Preserving the process of creation or the conditions of production is critical for any kind

of media, and malware is no exception. For an institution that wishes to collect material related

to the history of malware, any traces that indicate the process of creation should be saved if they

can be obtained; however, obtaining this information will be far more difficult for malware than

mainstream software. This documentation could answer the following questions: Was the

malware written by an individual, a group of several coders, a collective, a political organization,

or an organized crime ring? Was this malware created as a hobby or by contract?

Saving different iterations of source code, one can see the development of malware. In a

well-curated collection, a researcher could examine a single malware author's works over time to

trace their career. Many conventional documents that depict the production process in computing

such as "correspondence, working papers; unpublished reports; obsolete manuals; key program

listing used to debug and improve important software"[199] will be unavailable for malware. Nevertheless, archivists will have to be creative in how they work to capture the process of malware creation.

Careful Curation

In this chapter, I have tried to be as comprehensive as possible in suggesting material to collect for institutions wishing to create a malware archive. But malware collections can easily become enormous. Cultural heritage institutions can enhance malware collections with careful curation, historical perspective, supporting documentation, and intellectual linkages between items. One issue for cultural heritage institutions would not be the size (as most malware files are relatively small), but the complexity of the holdings. For example, CERT/CC's collection is so complex to search (since most of its files are binary) that its staff developed a special tool called "Big Grep" to look through all of the samples.[200]

Assembling and maintaining a collection of malware is no easy task. An archive seeking to achieve a "pure" collection of malware would benefit from asking for files from an already established collection of unique samples, such as that of CERT/CC or of a well-established virus researcher such as Vesselin Bontchev or Mikko Hypponen.

Bontchev's article "Analysis and Maintenance of a Clean Virus Library" chronicles the immense amount of time he has spent separating the wheat from the chaff when looking through public submissions of virus collections and individual viruses. Antivirus software is not much

---

[199] These items were recommended by the American Federation of Information Processing Societies. Quoted in Henry Lowood, "Shall We Play a Game: Thoughts on the Computer Game Archive of the Future," October 2002, http://web.stanford.edu/~lowood/Texts/shall_game.pdf.
[200] See https://github.com/cmu-sei/BigGrep

help in this regard as it can identify nonviruses, viruses that did not actually replicate, viruses that are incorrectly coded and cannot replicate, viruses whose replication requirements are far too specific so they need to be "spoonfed" to launch their payload, and snippets of virus code within text files. Bontchev must closely examine many files using different methods of analysis and software to determine if certain files are indeed unique viruses. Arguably, these non-virus items, which Bontchev considers trash, may have some historical value for a cultural heritage institution intentionally collecting malware and related artefacts. However, it would still be important for a cultural heritage institution to know which files are unique viruses.

"A grey area in virus collection is the so-called 'intended' virus. It is not really a virus because for some reason, programming error or compiling error, it malfunctions and does not replicate."[201] A cultural heritage institution that decides to collect only a limited number of viruses would likely skip intended viruses (unless they specifically related to a virus being collected), but an institution that wanted to preserve the phenomenon of malware programming more holistically would likely want to save intended viruses.

---

[201] Cicatrix, "Collecting Computer Viruses: Fun or Folly?"

Chapter 6: Metadata for Malware

The following chapter outlines metadata best practices and challenges either for an institution intentionally collecting malware or one that encounters malware-infected artefacts within its collection items.

Lack of Consistent Classification

At present, most malware is not classified consistently. For example, pieces of malware do not have the same name across different brands of antivirus software. Typically, antivirus researchers have named and classified malware, but "virus analysts, security experts, and computer scientists seem to stumble on disagreements and complications when it comes to the identification, the classification, and the naming of viruses. This is especially true when computer experts try to provide fast detection, real-time mapping, and elimination."[202] The quick detection and removal paradigm "prevents both expert and non expert from either classifying viruses...following consistent methods, or from properly distinguishing them according to their specific peculiarities (as worms, Trojan horses,[203] logic bombs,[204] etc.)."[205] When companies create antivirus software, the name of the specific piece of malware and even its classification (whether it is a virus, worm, backdoor, etc.) becomes only a minor detail. As a result, multiple antivirus companies have their own naming systems and may use a phrase from a line of the

---

[202] Parikka and Sampson, *The Spam Book*, 92–93.

[203] A Trojan Horse is a piece of malware disguised as something else. A Trojan Horse may be a benign-looking email, program, or file, but unleashes a payload once opened.

[204] A logic bomb is unwanted code that is inserted into a system and meant to be executed at a specific time, or under specific conditions. For example, a logic bomb integrated surreptitiously into the code of bank software might set all account balances to zero on April Fool's Day.

[205] Parikka and Sampson, *The Spam Book*, 92–93.

malware's code or from a payload screen as the malware's name. They may also use a common name that develops among computer users. On the other hand, Vesselin Bontchev suggests that one should avoid naming viruses after text found in the code because it "boosts the malware author's ego."[206] In fact, virus writers take great care in naming their creations: "Many virus authors name their viruses and if you believe the interviews with these authors many of them think it is one of the hardest parts of virus writing."[207]

A more scientific, sample-based naming scheme for malware has often been suggested, but "the problem with applying this approach in the anti-virus world has been the lack of a central reference collection or even a central naming body."[208] Thus, two companies may use different names for the same malware.

When antivirus companies started in the 1980s, there was no consensus on how to name malware. In 1991, CARO (Computer Antivirus Research Organization) developed a naming scheme, which has been continuously updated. The CARO scheme groups similar malware into families, gives a unique name for each sample, and has rules for naming variants. In a 2005 conference presentation, Bontchev continued to push for the adoption of the CARO naming scheme. Since CARO is only an advisory body, it cannot enforce decisions on naming conventions.[209] In addition, "Although most well known AV [antivirus] companies are members of CARO the virus naming habits of many of these companies have remained confusing and have never reached a CARO structure."[210] Some companies, such as Microsoft, use the CARO

---

[206] Bontchev, Vesselin. "Current Status of the CARO Malware Naming Scheme." presented at the Virus Bulletin Conference, Dublin, Ireland, October 13, 2005. https://www.virusbulletin.com/uploads/pdf/conference_slides/2005/Vesselin%20Bontchev.pdf.
[207] Cicatrix, "Collecting Computer Viruses: Fun or Folly?"
[208] Parikka and Sampson, *The Spam Book*. p. 92–93.
[209] Bontchev, "Current Status of the CARO Malware Naming Scheme."
[210] Cicatrix, "Collecting Computer Viruses: Fun or Folly?"

convention, but inconsistencies can still exist because "even when companies use a CARO-based naming scheme, the firms might differ in how they name identical malware specimens, for instance by using different approaches to selecting the Family Name component."[211] Another naming standard, the Common Malware Enumeration (CME) initiative was designed to create names for high-profile malware that the public could understand, but this effort is currently inactive.[212]

When it comes to naming malware, a cultural heritage institution would either have to forge its own path or accept the naming system of an existing company or organization. Of course, to enhance search capabilities, the records for any piece of malware would need to have linkages to both common and assigned names. Pieces of malware often have several aliases, for example, ILOVEYOU is sometimes referred to as a "worm," sometimes as a "virus," and is known as "Love Letter" or "Love Bug" and had several other names among members of the public (such as the "Love Virus") when it was originally released. In addition, ILOVEYOU has a huge number of variants, some written by unknown authors in several different countries, which send different messages in the infected email. Malware information aggregators such as Virustotal (which lists different antivirus software's names for a threat based on checksum) and ThreatExpert (which reports sightings of particular pieces of malware) could prove extremely useful to archivists when tracking information about malware names.

Understanding variants, copycats, and derivatives can shed light on other viruses. For example, the definitive creation date of the AIDS computer virus is unknown. However, "The time that AIDS was authored is estimated to be sometime closely before the time AIDS

---

[211] Zeltser, Lenny. "How Security Companies Assign Names to Malware Specimens." *Zeltser Security Corp.* Accessed April 24, 2016. https://zeltser.com/malware-naming-approaches/.
[212] Ibid.

derivatives were authored. The earliest known derivative of AIDS is Leprosy, authored in 1990.

Thus, AIDS is believed to be authored and isolated in early 1990."[213]

In attempting to make a searchable catalog of malware available to researchers, one must

also consider how variants are named by antivirus software versus how they may be commonly

understood by individuals outside of the computer security field:

> Since the origin of viruses does not have a high priority in naming computer viruses AV companies will group viruses with similar structures, layout or operating ways in families even though these viruses might have different authors and might have originated from different parts of the world. Variants of the Jerusalem virus might be given totally different names by their authors but if the only difference is found in the text contained in the virus an AV product will ID the viruses in the same family / group and will give them similar names.[214]

Classifying viruses in this way may prove entirely counterintuitive to a researcher who is seeking

information about viruses that originated in a certain country or were written by a certain author.

Researchers may be looking for a virus with a particular payload screen, but may be unaware

that it is a variant of another virus.

Antivirus programs work by detecting "signatures" or patterns in files that correspond to

particular pieces of malware. However, two variants with similar, but not precisely the same

code could have the same signature. If someone used the code from a preexisting worm and only

modified a few lines, the new variant could have the exact same signature, and an antivirus

program may not be able to indicate this to a user. A decompile and expert analysis of the code

may be required to determine if the virus is a unique variant.

Metadata about malware may now be harder to collect on the internet than in the recent

past. Computer security companies like F-Secure, McAfee, and Kaspersky Labs, who used to

---

[213] "AIDS (computer Virus) - Wikipedia, the Free Encyclopedia."
[214] Cicatrix, "Collecting Computer Viruses: Fun or Folly?"

have extensive information about malware on the internet are now removing it to make their sites more "consumer friendly."[215] Hopefully, some of this information has been saved through web archiving.

CERT/CC keeps extensive metadata about the malware it collects: "Some static (import hashes, function hashes, AV name, etc.), some dynamic (from the results of running in a sandbox, like files-dropped, registry keys added, domain names queried, etc.)." Edward Stoner at CERT/CC commented that the dynamic metadata is traditionally not kept for conventional software. He reasons that this is the case because "you wouldn't do it if you actually had the source code over time, and mostly someone does."[216]

Despite this existing metadata from computer security firms and security organizations, cultural heritage institutions will need to go further in collecting and creating metadata about a piece of malware and its relationships to associated explanatory or contextual material. Metadata about malware within a cultural heritage collection should not only consist of technical information, but also include historical information and relationships with associated documentation.

For an item considered "underground" or outré like malware, metadata will continue to be provisional or unknown (for example, information about its creator[s]). Creators could be groups or collectives without any specific indications of group identification. For example, some security analysts have pointed to inconsistencies in style between lines of the WANK Worm's code to argue that it was written by several individuals.[217]

---

[215] Vanhemert, "Watch 15 Awesome MS-DOS Viruses in Action."
[216] Stoner, "RE: Malware Preservation Research."
[217] See Dreyfus and Assange, *Underground*.

Documenting Removal

In some cases, avoiding the removal of malware when making it accessible to researchers may be impossible. An institution may also decide that keeping an infected copy of a hard drive is too burdensome. Even if both an infected and cleaned copy are kept, an institution may want to document the rationale for why a certain piece of malware was removed from a disk image and perhaps who made the decision—staff may even want to keep an ever evolving list of decisions on malware that is kept or removed.

As part of the evolving thinking about deposit agreements for born digital materials,[218] an institution may want to consider explicitly stating what their removal or retention policy is for malware. A policy on removal or retention (or removal for access) could also be stated explicitly in collecting policies or born-digital workflow documentation.

Intellectual Linkage and PREMIS

If an archive decides to save two versions of a hard drive or disk, there must be an intellectual linkage between both the "cleaned" and "infected" files or disk images. Otherwise, researchers may be misled into believing that the "clean" hard drive or disk is the only existing version, or they may not even realize that the artefact was infected in the first place. In addition, more detailed and standardized metadata about removal would inform both researchers and future archivists about a file's or drive's provenance.

Preservation metadata standards such as Preservation Metadata Implementation Strategies (PREMIS) could develop events for antivirus quarantine or removal. Ben Fino-Radin

---

[218] See, for example, Kirschenbaum et al., *Digital Forensics and Born-Digital Content in Cultural Heritage Collections*.

agrees that documentation of malware infections of born-digital artefacts is critical and believed

integrating documentation about malware removal into PREMIS metadata was a promising

research area.[219]

PREMIS records events that take place on objects (like files). "Virus scan" as an event is

already well established within the PREMIS standard, so events like "malware detection,"

"malware removal," or "malware quarantine" could also conceivably be created. Each event

would contain additional metadata including, but not limited to, the name of the malware (and

possibly the variant), the file path, and a timestamp. With these new events and the associated

metadata there would also need to be the creation or modification of software that can parse

these new PREMIS events and present them to an archivist or researcher so that records of items

in the collection could include the fact that a virus was detected and whether or not it was

removed. Software such as Archivematica could be modified to automate the process of writing

malware-related PREMIS events when its built-in antivirus software detects viruses. More

ambitiously, a different kind of antivirus software could be developed. This possibility will be

discussed further in Chapter 9.

For institutions that are intentionally collecting malware, the PREMIS environment

(which in PREMIS 3.0 can be its own independent object) can be used to capture all of the

hardware and software dependencies that the malware originally required. Information about

what was necessary to unleash the payload (e.g., a specific date or time) could also be recorded.

Since environments can be independent objects linked to a digital artefact through a dependency

relationship, institutions may want to create an environment object for both a "vintage"

---

[219] Fino-Radin, In conversation with the author.

environment and a contemporary environment. This could be especially helpful if malware

functions properly when run in certain emulators but not in others.

Each disk image will require its own metadata. However, in this area, there are more

robust standards and solutions: "Metadata associated with a raw stream will typically include

low-level hardware information such as drive geometry, system information from the capture

hardware, and cryptographic hashes to verify integrity."[220] Also important are "contextual links

between any captured drives and supporting materials...either specified during capture or prior to

ingest...information on image provenance and integrity should be recorded either via a packaging

mechanism (such as the 'case files' used in forensic investigations) or within formats specifically

designed to support the addition of flexible metadata [like AFF or E01]."[221]

David Rosenthal suggests the idea of including "usability metadata" in packages or

presentations of emulated software. Usability metadata, like information about keyboard and

mouse controls, helps users understand how to operate the software.[222] Usability metadata for

malware may cover the commands or the conditions necessary to run it or interact with it once

started. If the malware is triggered on a certain date, or by a certain series of actions, this

information should be included in its usability metadata as well.

Borrowing metadata from a computer security research firm or organization could assist a

cultural heritage institution in creating standardized metadata and faceted searching. Search

capabilities may have to go far beyond just the name of the malware and several aliases. For

example, the Big Grep search tool created by CERT/CC actually searches through the binary

---

[220] Woods, Lee, and Garfinkel, "Extending Digital Repository Architectures to Support Disk Image Preservation and Access," 61.
[221] Ibid.
[222] Rosenthal, "Emulation & Virtualization as Preservation Strategies," 15.

code. Other characteristics will also need to be searchable, such as transcriptions and

descriptions of payload screens.

Because of the obscurity of some malware, researchers may not know exact names for

what they are interested in. The only information some researchers may have about a piece of

malware is what it does. A database could enable this kind of searching through tagging

malware's payload with tags like "encrypts files," "deletes files," "alters .COM files," "locks

keyboard and/or mouse input," "installs backdoor," "installs keystroke logger," etc. Tagging by

operating system or by the type of malware (virus, worm, rootkit, ransomware, etc.) may also

help researchers find what they are looking for.

Chapter 7: Proof of Concept — Providing Access to Malware

Imagine an art exhibit of computer viruses. How would one curate such a show? Would the exhibition consist of documentation of known viruses, or of viruses roaming live in situ? Would it be more like an archive or more like a zoo...how would one curate an exhibit of disease? Would it include the actual virulent microbes themselves...would the epidemics have to be "historical" to qualify for exhibition?
—Alexander Galloway and Eugene Thacker, *The Exploit*

Like preservation strategies, access strategies for malware may need to be determined on a case-by-case basis and will depend on what an institution has actually acquired that relates to the individual piece of malware. Does the institution have the original source code or script? Compiled code? Disk images of infected hard drives? Logs from infected servers? Security advisories or news articles from when the malware was released? Email and online bulletin-board system (BBS) posts of people reacting to the infection? An extensive archive of websites related to the malware? Interviews with the malware's creator or people affected?

Just like other complex digital objects (such as computer-based artworks), understanding malware requires more information and context than simply inspecting an infected hard drive or lines of the malware's code. Examining ILOVEYOU's code will tell a researcher very little about how fast and far it spread (causing an estimated $5-8 billion worth of damage worldwide) or why it became such a media phenomenon (one of the most reported-on malware infections). This chapter presents and evaluates various methods for cultural heritage institutions to facilitate research on the history of malware.

Storing and Providing Samples

One method for storing malware, commonly employed by amateur virus collectors as well as computer security researchers, is to place infected files or files that are meant to be

infected (also called "goat files") in a disk image. In general, mounting an infected disk image on a live (not emulated or virtualized) computer is frowned upon by the computer security community. A test computer connected to a network may provide the malware with an additional means to spread.[223] Forensic disk image formats, such as the Advanced Forensic Format (AFF) and the EnCase Format (E01), have options to allow a user to examine files without actually mounting the image. Woods, Lee, and Garfinkel suggest that an AFF image "can serve as a kind of sandbox or staging area...to expose users to content from the image without requiring them to mount the drive or run the original file system. This can help to minimize risk of infecting users' computers with legacy computer viruses."[224] The AFF format has an API and application-level support for varying levels of access to a disk image that don't require mounting the image or providing a raw bitstream to the end user. In theory, an archive could allow different users to only look at specific sets of files.[225] Researchers with different goals and from different disciplines may want to examine different areas within the disk image. "A researcher examining the working environment and data creation practices of the user may be interested in the tools in the known files list; whereas an investigator analysing the impact of malicious software upon a live system would be interested in the known bad list."[226]

While AFF is still being maintained, "the community providing support for EWF (.E01 files) is much stronger." E01 offers a similar feature set to AFF including accessing streams of

---

[223] Further discussion on risk analysis will take place in Chapter 8.

[224] Woods, Lee, and Garfinkel, "Extending Digital Repository Architectures to Support Disk Image Preservation and Access," 58.

[225] Ibid., 63.

[226] Knight, "The Forensic Curator," 55.

disk images without mounting them and varying levels of encryption.[227] E01 appears to be a promising format for storing disk images which contain malware infections.

Tools for creating these kinds of images are already in the hands of archivists, librarians and conservators. Guymager disk imaging software is included in the BitCurator suite (a collection of free and open source software for digital preservation widely used by the cultural heritage community). Guymager allows for the creation of E01 as well AFF disk images.

Additional precautions are taken by virus researchers, who will often only look at static code. If viewing live malware becomes necessary, they will typically examine it behind tightly sandboxed emulated or virtualized environments on designated computers.[228] At CERT/CC several precautions are taken when investigating malware: "we mostly have Windows malware and mostly access it from Linux computers. Most of our analysis is static (meaning we don't have the malware run). We isolate systems whenever there is a need to have the malware actually run."[229] CERT/CC does provide researchers with samples of malware with associated metadata: "we may provide our own labeling (what we think the sample does), plus any tools we've developed for analysis."

The Internet Archive's Malware Museum

The Malware Museum, created by Mikko Hypponen and Jason Scott and hosted at the Internet Archive (IA), uses DOSBox emulation on IA's servers to present the payload screens of DOS viruses from the 1980s and 1990s.[230] The page for each virus contains minimal metadata

---

[227] Woods, "RE: Preserving Malware and EO1."
[228] Rosenthal, In conversation with the author.
[229] Stoner, "RE: Malware Preservation Research."
[230] See https://archive.org/details/malwaremuseum&tab=collection

and the "destructive routines" of the viruses have been removed within the emulation. Hypponen "overwrote replication and damaging parts [of the viruses] with the NOP ("No OPeration") command" and IA only has the crippled version of the virus.[231] Hypponen says that he removed the destructive routines for aesthetic reasons: "to make the animations and other visuals reproduce reliably and without delays."[232] He stated there were no legal or safety concerns about posting these pieces of malware. These historical viruses typically do not replicate or send data over computer networks and were likely spread through floppy disks or CDs.

Hypponen hopes to add more metadata to the Malware Museum when time allows. More information on these viruses would be extremely valuable. The current iteration of the Malware Museum does not delve into the inner workings of any of the viruses. For example, on the page for CRASH.COM, there is no additional information about the virus other than the emulation of the payload screen.[233] Daniel White's video on CRASH.COM explains that it infects .com files (executable files needed to run DOS), but nothing more.[234] Unfortunately, no additional information could be gleaned from the internet about how CRASH.COM spread or who wrote it.

In contrast to IA's Malware Museum, Daniel White's videos not only display payload screens, but often discuss the virus's history, what files on a user's computer were affected, what conditions triggered the payload, and how the payload affected the computer (such as locking keys or freezing the screen). While providing a widely accessible platform for the public, both of these projects have their limitations due to their reliance on a single or limited preservation or access strategy, whether the strategy is emulation or narrated screen recordings.

---

[231] Hypponen, "Re: Malware Museum and Malware Preservation."
[232] Ibid.
[233] *Malware Example: CRASH.COM*. MS-DOS, 2016. http://archive.org/details/malware_CRASH.COM.
[234] danooct1. *Crash.com DOS Virus*. Accessed March 12, 2017.
https://www.youtube.com/watch?v=vGRkfWea4HE.

IA's Malware Museum has demonstrated how to make (a limited form of) malware publically accessible through emulation with no risk of ill-effects to anyone's computer. However, despite using emulation, the format of the Malware Museum does not allow users to fully interact with the malware. The same is true of Daniel White's videos—viewers cannot interact with the infected computer. When speaking about video games, Henry Lowood states that "interactivity is about actions, not just content"[235] and the same could be said of malware—it may not be fully understood or appreciated without the user's ability to encounter the payload for themselves. They cannot experience the frustration of being locked out of the keyboard while the Phantom 1 virus runs or place their own bets against the Monte Carlo virus.

Access Through Public Exhibition

The digitalcraft project, begun in 2000 and based out of the Museum of Applied Arts in Frankfurt, addressed the issue of how to preserve born-digital artefacts of cultural production: "Digitalcraft has confronted the problem of building up a digital collection and has tested different solutions. The three collection areas now contain a selection of recent web design, games and emulators as well as a historical online community."[236] However, no staff members at the museum are currently working on the project and little remains of it online.

In the spring of 2002, digitalcraft mounted an exhibition of computer viruses and worms called "I Love You." Contributors to the exhibition's online catalog saw viruses as an important social phenomenon. In one essay, Massimo Ferronato argues that every virus sends a message

---

[235] Lowood, "Shall We Play a Game," 15.

[236] "About Digitalcraft: Projects & Concepts." *Digitalcraft.* Accessed May 2, 2016. http://www.digitalcraft.org/index.php?artikel_id=22.

about its creator and in saving malware one will be saving important information about those who wrote it.[237]

The exhibition itself featured installation of computers infected with ILOVEYOU and Biennale.py. There was also an area called "the zoo" where people could make their own viruses using virus creation tools and then have them run loose on dedicated computers. The "I Love You" exhibition opened at the Museum of Applied Arts in Frankfurt and subsequently traveled around Germany, then to Providence, Rhode Island, and then to several other locations in Europe. Jussi Parikka states that the museum may have added computer viruses to its collection; [238] however, I have not been able to get in contact with the digitalcraft project or the Museum of Applied Arts to determine if the museum did indeed collect malware. Digitalcraft's website states that exhibition materials were kept.[239] If malware was accessioned, digitalcraft was perhaps the first cultural heritage project to attempt to preserve malware.

The "Project Cyber Virus" exhibition, held in 2015 at Swissnex San Francisco, explored viruses from the 1980s and 90s. Swissnex is a cooperative venture between Switzerland and countries in North America. At the opening reception, virus collector Daniel White performed live demonstrations of malware. Both "I Love You" and "Project Cyber Virus" widely emphasized interactive components.[240]

---

[237] Ferronato, Massimo. "The VX Scene." *Digitalcraft*. Accessed May 2, 2016. http://www.digitalcraft.org/?artikel_id=285.

[238] Parikka, *Digital Contagions,* 285.

[239] Nori, Franziska. "A Decade of Web Design." *Digitalcraft*, January 2005. http://www.digitalcraft.org/index.php?artikel_id=550.

[240] swissnex San Francisco. "Project Cyber Virus: Digital Security Then and Now." *Swissnex San Francisco*. Accessed January 5, 2016. http://www.swissnexsanfrancisco.org/event/projectcybervirusexhibit/; swissnex San Francisco. "Project Cyber Virus: Opening Reception." Swissnex San Francisco. Accessed January 5, 2016. http://www.swissnexsanfrancisco.org/event/cybervirusopening/.

Many innovative possibilities still exist for publicly exhibiting malware. In *The Exploit*, Galloway and Thacker propose the concept of an ever-evolving exhibition: "the exhibit would require the coordination of several museums, each with 'honeypot'[241] computers...A network would be required, the sole purpose of which would be to reiterate sequences of infection and replication."[242]

Importance of Accessible Primary Source Materials

Considerable secondary source material often exists about malware infections, including books, newspaper articles, and online content. However, without sufficient primary source material that provides authoritative information about the malware itself (such as code), it may be impossible to verify information contained within these secondary sources. Particularly at a time of crisis, press releases, newspaper articles, and even firsthand testimony can become exaggerated. This makes access to primary-source materials exceptionally important as a means to assess past statements and research.

Suelette Dreyfus, who researched the WANK Worm along with Julian Assange, used government memos and posts on security bulletin boards when writing *Underground*. She also had access to an archived mailing list from VMS-system security administrators and a version of the code. Dreyfus said that she did not use libraries or archives to conduct her research on the worm.[243] This is likely because at the time the book was written, 1997, these institutions would have had little information of use to her. However, if no cultural heritage institution accepts the

---

[241] A honeypot computer is intended to attract hackers or malware. The computer is used as bait and is usually closely monitored to collect evidence against individuals who launch attacks against it. A honeypot could also simply be dummy data on a website that appears attractive to attackers.
[242] Galloway and Thacker, *The Exploit,* 105.
[243] Dreyfus, Suelette. "RE: Research on WANK," March 16, 2015.

challenge of preserving malware and related contextual material, researchers like Dreyfus may

lose the ability to conduct similar analysis in the future.

Purpose-Built Computers and Emulation

Some researchers may be satisfied by just looking at static code, but for those who want

access to infected hard drives to study malware infections and their effects in more depth, the

situation may become more complicated. For those who want to browse the files on an infected

hard drive, a set of files from a forensic disk image loaded into a virtual machine or emulator

may suffice. Researchers who want to see malware demonstrated will be the toughest

customers—highly contained computer systems purpose-built for demonstration that run

malware in a virtual machine (VM) or emulator appear to be the safest course of action. If

malware is to be demonstrated, the specific hardware, operating system, and software it exploited

must be preserved, potentially in an emulated or virtualized environment.

The use of emulation or virtualization comes with its own set of complications and

quirks. Even if it works reasonably well, emulation is not always a perfect translator of software

and can change its aesthetic qualities or intended behavior; it remains to be seen if emulators can

consistently run specific pieces of malware as their creators intended.[244]

Several other issues with emulation still remain unresolved. "Concern about the level of

support for the emulators needed for preservation was universal" among the institutions that

David Rosenthal consulted with for his study on emulation and virtualization, including the

---

[244] On the translation problem of emulation see Besser, Howard. "Longevity of Electronic Art." Accessed May 7, 2015. http://besser.tsoa.nyu.edu/howard/Papers/elect-art-longevity.html; Besser, Howard. "Digital Longevity." Accessed August 20, 2017. http://besser.tsoa.nyu.edu/howard/Papers/sfs-longevity.html.

Internet Archive, Rhizome, the British Library, and others.[245] Rhizome's digital conservator

Dragan Espenschied has written that "emulation development is either driven by hobbyists or the

needs of big business...There is a lot of overlap with the interest of cultural and memory

institutions, but in general, everybody seems to be working with by-products."[246]

Almost all malware is parasitical on commercially available software. However, the legal

framework of running commercial software under emulation (even if the archive has legally

purchased a copy), is somewhat unclear or highly restrictive. "Institutions generally lack a clear

understanding of exactly what rights they acquired when they purchased commercial software

licenses, whether the rights cover execution in emulators and VMs."[247] This problem will affect

the future of emulation, which live malware demonstrations will depend on.

As malware requires specific operating systems to function, an institution may have no

choice but to purchase niche emulation or virtualization software for demonstration purposes.

For example, to run the WANK Worm, Garfinkel suggested an archive purchase a VAX

emulator called CHARON-VAX developed by Stromasys, but also noted that the costs of such a

system may be substantial.[248]

As a study by Hedstrom et al. confirms, simply providing a researcher with an emulated

version of software is not enough. Users of emulated software needed information about the

original computing environment: "this type of contextual information will become even more

important as users, over the course of time, become less and less likely to have had firsthand

knowledge or experience with obsolete computing platforms."[249] For example, few young adults

---

[245] Rosenthal, "Emulation & Virtualization as Preservation Strategies," 13.
[246] Quoted in Rosenthal, "Emulation & Virtualization as Preservation Strategies," 13–14.
[247] Rosenthal, "Emulation & Virtualization as Preservation Strategies," 16.
[248] Garfinkel, "Re: Research on Malware."
[249] Hedstrom et al., "The Old Version Flickers More." p. 187

have experience using a computer outside of a graphical user interface, or without a mouse; almost no person born in the 2000s has experience using MS-DOS as an operating system. In this study, the researchers concluded that "three types of contextual information were particularly critical for the subjects in our experiments: information about the context in which the objects were originally created and used; information about the purpose and audience for the materials; and information about the original computing environment."[250] Detailed contextual information about how the digital objects were created related to particular pieces of malware may not be available, but cultural heritage institutions that intentionally collect malware should strive to locate it.

While certainly less risky than running malware on a live computer, using emulation and virtualization still presents risks to the safety of networks and computers. When running any older software, undiscovered vulnerabilities will always exist: "the interval between discoveries of new vulnerabilities in released software *decreases* through time. Thus the older the preserved system image, the (exponentially) more vulnerabilities it will contain."[251] The next chapter will outline avenues for assessing the risk of storing and providing access to malware.

---

[250] Ibid.

[251] Rosenthal, "Emulation & Virtualization as Preservation Strategies," 23.

Chapter 8: Risk Assessment Considerations for Storage and Access

An archivist, conservator, or librarian must take a sanguine view about storing malware in a digital repository and allowing access to it. The risks that malware accessioned into a collection pose to a cultural heritage institution may be generally classified into four distinct categories:

1. Introducing risks to the integrity of other files in a digital repository (as malware can delete or modify files).

2. Allowing unauthorized access to the repositories or other computer systems of an institution. Malware frequently creates backdoors, which could be exploited by unauthorized users who may either modify or delete files from a digital repository, use the backdoor to launch attacks against the institution, or use the institution's machines as part of a botnet to send spam, launch attacks on other computers, or spread malware further.

3. Posing legal liability risks to the archive. This is related to the previous risk factor. Since some malware makes network connections, it could affect computers outside the institution. If other people's computers are harmed as a result of malware being run by a museum or archive, these people could hold the archive accountable. Malware collections could potentially pose legal liability risks in a different manner. If a researcher took the malware out of the archive and started using it against others, in theory, the institution could be held liable. David Rosenthal believes that legal liability risks may pose a huge impediment to institutions

actively collecting malware.[252] These risks can be mitigated by taking measures outlined later in this chapter.

4. Preventing or inhibiting researcher access to data on a digital artefact. To demonstrate how malware can inhibit access, Jane Gruning gives the example of floppy disks that were infected with a boot-sector virus called Stoned, where the infection prevented files on the disks from being read properly.[253] If the malware's payload is a particularly obtrusive screen, or continual freezing or crashing of the operating system, this would also compromise access to an infected drive or computer if the researcher's intent was to examine the entire computing environment and not just the malware infection.

In assessing risk, one may also want to separate malware into two classes—malware that makes network connections and malware that does not. This attribute dramatically affects the risks incurred during access. For example, when accessing MS-DOS viruses that do not make network connections, setting up a sacrificial computer to demonstrate viruses is not out of the question since it would be impossible for the malware to spread to another computer without intentionally copying it. Daniel White has used a designated computer to create his malware videos for years. Clearly, malware that does not make network connections poses less risk, but the vast majority of post-2000s malware makes network connections.

Malware Within Disk Images

---

[252] Rosenthal, David S. H. In conversation with the author. Phone call, April 19, 2016.

[253] Gruning eventually had to extract a copy of the original bitstream by using a hex editor. In comparing code from the cleaned disk and the infected disk, she was able to isolate the code of the virus. She states that "the original bitstream of the infected disk image has been retained and cataloged with the rest of the collection although it is not available to the public." Gruning, Jane. "Rethinking Viruses in the Archives." Poster presented at the Archival Education and Research Institute, 2012. https://www.ischool.utexas.edu/~janegru/images/Gruning_AERI2012.pdf.

There are two points of risk to consider for malware within disk images: when the image is initially created, and when the image is accessed. As Christie Peterson learned, even conducting a virus scan at the imaging stage can trigger certain kinds of malware payloads. However, while the disk image remains unopened the malware is basically inert. Writing the disk image to an LTO tape (as opposed to the hard drive of a computer) provides an additional level of segregation between the malware and a live computer system.

For institutions preserving malware, opening infected disk images requires a risk assessment. Mounting disk images onto a computer's file system poses potential risks: "Accessing disk images via a host mount imposes a number of technical limitations...the researcher...may incur security risks on the host due to virus infections present on the imaged system."[254] For example, some versions of Windows may automatically run a file called "autorun.inf" when the image is mounted. If so, opening the image could launch a program or script activating malware. As previously discussed, AFF and E01 disk images do not need to be mounted in order to be examined, and specific streams of data can be designed for researchers who want access to the malware and infected files and those who do not.

Operating System Risks

In the case of malware, obsolescence is actually a double-edged sword. On the one hand, obsolescence makes demonstrating malware less straightforward when using contemporary computer systems. On the other hand, malware intended for obsolete systems may be completely harmless to contemporary computers, and the malware can be examined in a static way without

---

[254] Woods, Lee, and Garfinkel, "Extending Digital Repository Architectures to Support Disk Image Preservation and Access," 59.

posing much risk. "Many viruses make assumptions about hard-coded addresses or undocumented structures in the operating system - and are therefore limited to a particular version" or only replicate on a machine with a certain CPU or a specific amount of RAM.[255]

Malware is often specific to an operating system or even a particular program. For example, some malware is packaged as a Windows executable file, which cannot run natively on macOS. This means macOS is immune to the particular piece of malware, which would simply be stored as data. Static analysis of the Windows malware (reading or decompiling the code) could occur on a macOS or Linux computer without any risk of activating its payload. The Linux operating system appears to be the best choice for static analysis as it has far fewer threats than the other major operating systems.[256] CERT-CC conducts its static analysis on Linux machines.

Microsoft has tried to maintain a high degree of backward compatibility with its Windows operating system, which means that malware intended for Windows 95 could theoretically run on newer versions of Windows. There are many differences between versions of Windows, like registry keys, the locations of various configuration files, and drivers. Thus, a deep level of analysis would be needed to determine the risks of opening an older infected hard drive on newer versions of Windows.[257]

While much malware is operating system dependent, some malware takes advantage of cross-platform frameworks (such as Adobe Flash, Java, Python, or JavaScript) to infect computers regardless of the operating system. As long as a computer is capable of opening a Flash file, it could be susceptible to malware associated with Flash.

[255] Bontchev, "Analysis and Maintenance of a Clean Virus Library"
[256] Rovelli, Paolo. "Don't Believe These Four Myths about Linux Security." *Sophos News*, March 26, 2015. http://news.sophos.com/en-us/2015/03/26/dont-believe-these-four-myths-about-linux-security/.
[257] Chiu, Jeff. "Malware Preservation," April 26, 2015.

Emulation and Virtualization Risks

Any live demonstration of malware should ideally confine itself to an emulator or virtual machine (VM). The possible effects of running the malware in this manner will vary depending on the parameters set by the virtualization or emulation software. A tightly sandboxed VM may be capable of running the malware and containing its payload within the VM's environment without affecting the host computer.

Demonstrating or viewing malware that makes network connections can be a catch-22 within emulated or virtualized environments. If the VM does not allow network connections, the malware may not execute properly for demonstration purposes. However, if the VM allows network connections and is linked to a local area network or to the internet, it offers no protection, and the malware could spread. David Rosenthal suggests that "an important if minimal pre-condition for making emulation safe for networked digital artefacts is the development of encapsulation techniques for Internet Emulators capable of preventing emulators of old software being compromised in ways that affect other systems."[258] Such a development would allow institutions to run malware that makes network connections in a safer manner. Network encapsulation techniques and the practice of simulating networks to study malware are both common practice and areas of active research for the computer security community. Researchers increasingly try to simulate more convincing networks to bait the malware they are studying. In a 2012 paper about using network containment to analyze malware, the researchers examined "protocol learning techniques for the emulation of the external network environment."[259]

---

[258] Rosenthal, "Emulation & Virtualization as Preservation Strategies," 25.

[259] Mariano Graziano, Corrado Leita, and Davide Balzarotti, "Towards Network Containment in Malware Analysis Systems" (ACM Press, 2012), 339, doi:10.1145/2420950.2421000.

The level of risk when running malware within an emulator or VM may also depend on how the virtual hard drives are configured. If the virtual hard drives are not sealed off from the file system of the host machine this may be a tunnel through which the malware could infect the host machine.[260]

In addition, various VMs and emulators have their own sets of vulnerabilities and exploits. The most serious vulnerabilities allow an attacker to get outside of a VM or emulator and target the host system. An exploit called Venom existed in the floppy drive code used by many VMs and allowed an attacker to escape the virtual environment. Venom was publicly disclosed and has now been patched by the majority of major VM developers, yet this vulnerability existed in the code of VMs for eleven years.[261] In 2015, Rosenthal noted that "five significant vulnerabilities have been discovered so far this year" for QEMU, a popular open-source emulator and virtualizer.[262]

For reasons of fidelity to the original system, emulation programs may allow vulnerabilities to remain. The Olive emulation project believes the original software's vulnerabilities are "an essential part of the emulation, as they may themselves be the object of study." The project also notes that "other [emulation] frameworks adopt the same position."[263] Nonetheless, the Olive emulation project has a firm stance on the ethics of using an emulator to run malware designed to use the victim's computer to attack others: "even if the user of the emulation [such as a cultural heritage institution] were untroubled by the compromise of their

---

[260] Chiu, "Malware Preservation."

[261] "VENOM Vulnerability." Accessed April 28, 2016. http://venom.crowdstrike.com/; Korolov, Maria. "Significant Virtual Machine Vulnerability Has Been Hiding in Floppy Disk Code for 11 Years." CSO Online, May 13, 2015. http://www.csoonline.com/article/2921589/application-security/significant-virtual-machine-vulnerability-has-been-hiding-in-floppy-disk-code-for-11-years.html.

[262] Rosenthal, "Emulation & Virtualization as Preservation Strategies," 13.

[263] Quoted in Rosenthal, "Emulation & Virtualization as Preservation Strategies," 25.

emulated system, it would not be ethical to allow their emulated system to attack others."[264] A cultural heritage institution could not allow malware to run in a network-connected emulator or VM for both ethical and liability reasons.

These differing possibilities and scenarios require that the cultural heritage community both "develop archival standards for storing viruses safely,"[265] as Jane Gruning suggests, and develop standards for risk assessment and access. Ideally, an institution would provide a dedicated computer for access, but if this is impossible and researchers are using their own computers, "hosting institutions should ensure users are aware of its [malware's] presence and can take steps if necessary to protect their PCs."[266] In this case, having sufficient documentation of malware becomes particularly important so the institution can advise the researcher appropriately.

As malware gets stronger and more sophisticated (particularly worms similar to Stuxnet) it starts to take on the dimensions of a destructive weapon. With the code of Stuxnet freely available on the internet, one can imagine a programmer modifying Stuxnet's code to target programmable logic controllers connected to water systems or electric grids.[267] Preserving older malware for obsolete computer systems may be more akin to preserving Revolutionary War-era muskets, while preserving the sophisticated and destructive malware of the future may more closely resemble preserving information about constructing nuclear weapons.

---

[264] Ibid.
[265] Gruning, "Rethinking Viruses in the Archives."
[266] Pennock, "Web Archiving," 14.
[267] "Rise of the Hackers." *NOVA*. Accessed April 20, 2016.
http://www.pbs.org/wgbh/nova/tech/rise-of-the-hackers.html.

Antivirus Software

Consumer antivirus software uses several heuristic strategies to detect malware.

Traditional antivirus software uses signatures which are:

> small byte patterns that correspond to malicious code. Since only already known malware (where a signature have been created) can be detected, this approach requires a fast response to new malware. As long as there is no signature for a certain malware it will pass through the virus scanner as a non-malware file. The signatures have to be carefully chosen, so that they are not found in innocent files by coincidence. There is always a trade-off between fast generation of signatures and avoiding false positives.[268]

In previous decades, antivirus companies have had the luxury of analyzing malware closely to create signatures, but with the amount of malware increasing exponentially, automated methods of creating signatures had to be devised. Consumers download the latest signatures from the antivirus company and the virus scanner examines their hard drive to find any matches. As previously mentioned, viruses and worms have methods to hide their signatures through polymorphism. In order to combat polymorphic viruses, two heuristics were recently developed: sandbox detection (running a program within a tightly controlled environment, which is usually resource intensive and is rarely used in consumer antivirus software) and data mining (attempting to detect patterns in code being executed or changes to files to determine if a computer has a malware infection). However, the virus scanning software on most personal computers still uses signature detection, which means some malware could escape detection. In addition, "AV software typically errs on the side of caution and is known to result in false positives, so exclusion of material on the basis of a positive AV software scan could result in unnecessary omissions."[269]

---

[268] Ask, Karin. "Automatic Malware Signature Generation," October 16, 2006.
http://www.gecode.org/~schulte/teaching/theses/ICT-ECS-2006-122.pdf, 1.
[269] Pennock, "Web Archiving," 14.

Malware presents complications to any risk assessment: even if an anti-virus firm or an

individual has developed a patch to remove or quarantine malware, its complete effects may not

be well-understood. This type of software analysis—describing everything that a piece of

software does—requires a highly skilled analyst and is time consuming.

A temporary solution to reduce risk involves focusing solely on the preservation of older

malware. By preserving older malware, the institution will benefit from some historical

perspective. Unfortunately, if institutions are overly cautious and do not capture contemporary

malware in a timely manner, the malware's code and its associated documentation may disappear

forever. As a starting point, institutions can begin collecting malware that is historically

significant and well-understood and whose effects are relatively innocuous, and then move on to

more challenging cases as they gain more experience (similar to the path the Internet Archive has

started on with the Malware Museum). They could also start collecting documentation and

ancillary materials related to current malware now to better understand the malware when they

actually accession it in the future. There are no simple answers to the potential risks involved in

attempting to preserve and provide access to malware and many decisions regarding its

collection or preservation may require a case-by-case evaluation.

A Viral Dark Archive

Jane Gruning suggests storing potentially dangerous malware on limited-access,

non-networked storage—what she calls "a dark archive." She further suggests a quarantine

period for storing contemporary malware until the systems it runs on become obsolete.[270] While

---

[270] Gruning, "Rethinking Viruses in the Archives."

these suggestions are prudent, they may be difficult to implement. Unfortunately, there is no predictable timetable for when individual pieces of malware become less dangerous, particularly if their full capabilities are unknown. Some vulnerabilities that were made public many years ago still have not been fixed and for every piece of software some potential vulnerabilities will always remain unknown.[271] As Rosenthal points out, "once a vulnerability is exploited it becomes a semi-permanent feature of the Internet. For example, the Conficker worm appeared in November 2008...In mid-2011 Microsoft was still detecting 1.7M [million] infections each quarter."[272]

Malware that makes connections to the internet almost certainly has too many potentially risky outcomes and too many liability concerns to run on internet-connected computers. The potential for malicious attacks originating from the computer systems of a cultural heritage institution would not only be unethical and damage the institution's credibility, but could also invite lawsuits. Other computers connected to the internet do not always have the latest software patches and could be vulnerable to the malware being demonstrated at an archive. Because of these concerns, when demonstrating malware, CERT/CC uses a "custom sandbox" and an "internet emulator" and never runs malware "in a way that it could connect to the actual internet."[273] Any cultural heritage institution interested in demonstrating malware that makes network connections would be wise to follow CERT/CC's lead.

As computer security researchers have proven, demonstrating malware within a controlled environment so that one can conduct a fine-grained analysis is definitely possible.

---

[271] See, for example, Stockley, Mark. "The Web Attacks That Refuse to Die." Accessed March 13, 2017. https://nakedsecurity.sophos.com/2016/06/15/the-web-attacks-that-refuse-to-die/.
[272] Rosenthal, "Emulation & Virtualization as Preservation Strategies." p. 23
[273] Stoner, "RE: Malware Preservation Research."

Interested institutions ought to review the literature on malware analysis to gain practical insight into how to setup and configure dedicated computers or networks, which allow researchers to experience the malware while posing little to no risk to the institution or the general public.

Chapter 9: Further Questions and Research

Internet viruses might prove to be a micropolitical counter power, shaking majoritarian notions of software and the order of digital culture. Similarly they can be grasped as philosophical and artistic machines that create new perceptions and concepts.
—Jussi Parikka, "Archives of Software: Malicious Code and the Aesthesis of Media Accidents" in *The Spam Book*

This final chapter discusses avenues for further research, discussion, and action regarding the preservation of malware and infected digital artefacts. As cultural heritage institutions continue to collect more personal computers, floppy disks, optical discs, and web servers, the probability of encountering malware-infected hard drives or disks becomes higher. At some point, institutions can no longer afford to consider infected digital artefacts as anomalies and treat them according to ad hoc solutions.

In order to pinpoint the practical issues at hand, conducting a detailed national or international survey of how cultural heritage institutions handle malware-infected artefacts seems like a prudent first course of action. Armed with this information (released as a report or a journal article), the discussion among archivists, librarians, and conservators can begin in earnest. Through specialized symposia and presentations or panels at conferences or other events, cultural heritage professionals can help build community consensus on concrete workflows and best practices for handling malware-infected digital artefacts. By necessity, the discussion must be cross-disciplinary with computer security researchers, digital forensics experts, and digital studies scholars all engaging with cultural stewards.

Institutions that do not intentionally collect malware must keep in mind how the infection contributes to the context of their acquisition. Literature should be published that helps archivists understand what they are losing if they remove any infection.

Due to the archival issues previously discussed about antivirus software—namely, that it modifies data during virus removal or quarantine—the development of antivirus software for archives is a topic for further discussion and research. Perhaps antivirus software can be developed that is the equivalent of the Japanese tissue paper used in paper conservation. The tissue binds tears in paper documents to strengthen the item for handling, but clearly looks like a repair. In a similar manner, archival antivirus software could keep a record of exactly what files were affected; if applicable, how they were altered; and what piece of malware was found on the drive in a report that would be understandable to a researcher, or could easily be incorporated into the institution's catalog. Japanese tissue paper repairs are also fully reversible; in other words, the tissue paper can be completely removed without affecting the original document in any way. Similarly, archival antivirus software should include reversibility as a feature. If a malware infection is removed or quarantined, enough data would be saved so that reversing the removal or quarantine would be possible.

Another area of suggested research involves using private and secure collection methods for malware. Working with malware creators to collect and preserve current malware could inadvertently aid law enforcement if precautions are not taken. An archive could utilize cryptographic methods and anonymous networks, similar to the way Wikileaks receives data while protecting sources. The institution could also develop anonymous donation procedures and use strong encryption (though encrypting data creates its own archival difficulties) and delete any identifying logs.[274] However, as the subpoenas issued for materials in Boston College's Belfast Project reveal, an archive will never be able to guarantee the continued secrecy of the

---

[274] Though this method of collection may complicate the provenance of the donation. Vigorous debates ought to take place to help define archival standards for the donation of malware.

transaction or the anonymity of the donor. This raises the question of whether a malware coder would ever willingly donate their "papers" to a collection.

Developing archival donor agreements that involve infected hard drives or disks are another area of future work. Archives may also want to ask donors if they are willing to donate their hard drives toward malware research. In other words, if there were a central institution collecting malware, the hard drive could be shipped to the malware-collecting institution and then imaged by their staff to preserve an example of a particular malware infection "in the wild." Perhaps in the future, an institution would be interested in collecting the hard drives of "average people" who were infected (in some sense, the equivalent of film collectors who visit thrift stores looking for home movies and discarded film collections).

Copyright law affects how a cultural heritage institution can preserve, migrate, or allow access to computer code, as creators of software have intellectual property interests in that software. In most cases, copyright for malware may be a non-issue, as most creators of malware purposely do not want to identify themselves. Collecting and preserving malware code may elicit intellectual property claims or lawsuits in the future. Archives that are prepared to intentionally collect viruses can begin to discuss intellectual property auditing for malware code. For viruses intentionally created by artists who identify themselves, even those that are publically available (like Biennale.py), more attention to intellectual property concerns should be paid.

Potential liability and legal issues can be explored as well. Though the intentional collection or preservation of malware by a cultural heritage institution is likely to be legal, institutions should be prepared to defend their collecting policies. Cultural heritage institutions who operate for educational purposes ought to have as much of a defensible position as antivirus

companies or organizations like CERT/CC that collect and preserve malware. The question may hinge on the level of accessibility and what pieces of malware are made accessible. For example, institutions may have to have specific procedures for alerting researchers about the presence of malware, or not provide malware-infected artefacts unless specifically asked. This raises the following question: if it is impossible to give a researcher full access to a malware-infected digital artefact, what constitutes authentic access? The answer will have to be determined through professional discussion.

Institutions intentionally collecting malware will have to develop policies around providing samples. Otherwise, these institutions potentially face liability concerns if someone uses a sample for destructive purposes. Perhaps the institution determines some minimal bars that a researcher must clear before they have access to these samples. Or perhaps there is a hierarchy of samples with different levels of access based on the impact of a potential infection. Older or less dangerous malware would be more accessible.

We must return to the question of what constitutes a historically representative sample of malware. How does one go about defining the contours of the history of malware? After all, the histories delineated may end up determining archival collecting policies. In the best case scenario, collections can branch out like a tree with some collecting policies very different from others. What criteria are important in shaping a collecting policy for malware? Should some archives focus on collecting samples displaying coding innovation, others on creative expression, others on the assumed political intentions of the coder, and perhaps still others collect in an encyclopedic manner (attempting to collect in every defined area)? Do other more appropriate criteria exist?

Malware with new methods of infection will inevitably appear and "as newer security operations are developed to confront the spread of viral code, proposing new strategies that could possibly anticipate next-generation viral attacks, the reactions of virus writers will follow in the form of new viral agents aimed at shattering newly built security shields."[275] In preserving this new malware of the future, archivists may then have to radically rethink any current preservation paradigm. Finally, large-scale coordinated online/offline protests continue to raise questions about how to preserve the software involved in these events, along with its context and the social interactions it engendered. This speaks to an issue raised earlier, i.e., how to preserve lived experience.

None of the issues that the development of a malware archive raises are easy to solve, but neither were the challenges of collecting the history of cinema, television, or time-based media art.

---

[275] Parikka and Sampson, *The Spam Book*, 90.

Acknowledgements

Sources Consulted

"About Digitalcraft: Projects & Concepts." digitalcraft. Accessed May 2, 2016.
http://www.digitalcraft.org/index.php?artikel_id=22.

"AIDS (computer Virus)." Wikipedia. Accessed May 2, 2016.
https://en.wikipedia.org/wiki/AIDS_(computer_virus)#cite_note-1.

"Antivirus Software." Wikipedia. Accessed May 1, 2016.
https://en.wikipedia.org/wiki/Antivirus_software#2005_to_present.

Ask, Karin. "Automatic Malware Signature Generation," October 16, 2006.
http://www.gecode.org/~schulte/teaching/theses/ICT-ECS-2006-122.pdf.

Assange, Julian. "The Curious Origins of Political Hacktivism." Counterpunch, November 25,
2006. http://www.counterpunch.org/2006/11/25/the-curious-origins-of-political-hacktivism/.

Avram, Chris. "Re: Research on Malware," March 2, 2015.

Aycock, John. "Stux in a Rut: Why Stuxnet Is Boring." Virus Bulletin, September 1, 2011.
https://www.virusbulletin.com/virusbulletin/2011/09/stux-rut-why-stuxnet-boring.

"Back Orifice - Malware - McAfee Labs Threat Center." McAfee Labs. Accessed April 25,
2015.
http://www.mcafee.com/threat-intelligence/malware/default.aspx?id=10002&region=us.

Bailey, Jefferson. "Re: Viruses and Malware," May 3, 2015.

Berry, David M. *The Philosophy of Software: Code and Mediation in the Digital Age*.
Basingstoke, Hampshire ; New York: Palgrave Macmillan, 2011.

Besser, Howard. "Digital Longevity." Accessed August 20, 2017.
http://besser.tsoa.nyu.edu/howard/Papers/sfs-longevity.html.

———. "Longevity of Electronic Art." Accessed May 7, 2015.
http://besser.tsoa.nyu.edu/howard/Papers/elect-art-longevity.html.

Bontchev, Vesselin. "Analysis and Maintenance of a Clean Virus Library." VX Heaven, 1993.
http://vxheaven.org/lib/avb01.html.

———. "Current Status of the CARO Malware Naming Scheme." presented at the Virus
Bulletin Conference, Dublin, Ireland, October 13, 2005.
https://www.virusbulletin.com/uploads/pdf/conference_slides/2005/Vesselin%20Bontchev.pdf.

Brunton, Finn. *Spam: A Shadow History of the Internet*. Cambridge, Mass.: MIT Press, 2013.

Chiu, Jeff. "Malware Preservation," April 26, 2015.

Cicatrix. "Collecting Computer Viruses: Fun or Folly?," March 1999.
http://vxheaven.org/lib/static/vdat/epcolvir.htm.

Coleman, Gabriella. "The Public Interest Hack." *Limn* (blog), May 9, 2017.
http://limn.it/the-public-interest-hack/.

"Computer Virus." Wikipedia. Accessed May 13, 2015.
http://en.wikipedia.org/wiki/Computer_virus.

"Computer Worm." Wikipedia. Accessed May 13, 2015.
http://en.wikipedia.org/wiki/Computer_worm.

"Conan." TBS, May 13, 2014. http://teamcoco.com/video/george-r-r-martin-dos-program.

"Conservation (cultural Heritage)." Wikipedia. Accessed May 13, 2015.
https://en.wikipedia.org/wiki/Conservation_(cultural_heritage).

Critical Art Ensemble. "Electronic Civil Disobedience," 1994.
http://www.critical-art.net/books/ecd/ecd2.pdf.

danooct1. *Crash.com DOS Virus*. Accessed March 12, 2017.
https://www.youtube.com/watch?v=vGRkfWea4HE.

———. "Email-Worm.Win32.Loveletter (ILOVEYOU Worm, 12 Years Later)." YouTube, May
4, 2012. https://www.youtube.com/watch?v=ZqkFfF5kAvw.

———. "Virus.DOS.AIDS." YouTube, December 23, 2011.
https://www.youtube.com/watch?v=tckwz0ZS3Zo.

———. "Virus.DOS.MonteCarlo." YouTube, December 17, 2015.
https://www.youtube.com/watch?v=8mwZiGbO0Xc.

"Darkode." Podcast. *Radiolab*, September 21, 2015. http://www.radiolab.org/story/darkode/.

Davis, Joshua. "Hackers Take Down the Most Wired Country in Europe." WIRED, August 21,
2007. https://www.wired.com/2007/08/ff-estonia/.

Dreyfus, Suelette. "RE: Research on WANK," March 16, 2015.

Dreyfus, Suelette, and Julian Assange. *Underground: Tales of Hacking, Madness, and Obsession on the Electronic Frontier*. Kew, Australia: Mandarin, 1997.

Dumitras, Tudor, and Petros Efstathopoulos. "The Provenance of WINE." Symantec Research Labs, n.d. https://users.ece.cmu.edu/~tdumitra/public_documents/dumitras12wineprovenance.pdf.

epidemiC. "Biennale.py." Accessed May 1, 2016. http://epidemic.ws/biennale_press/01.htm.

"Equation Group." Wikipedia. Accessed December 6, 2015. https://en.wikipedia.org/wiki/Equation_Group.

"Exhibit Features Viruses as Art." WIRED, August 27, 2014. http://archive.wired.com/culture/lifestyle/news/2004/08/64724?currentPage=all.

Ferronato, Massimo. "The VX Scene." digitalcraft. Accessed May 2, 2016. http://www.digitalcraft.org/?artikel_id=285.

Finley, Klint. "Pro-Government Twitter Bots Try to Hush Mexican Activists." WIRED, August 23, 2015. https://www.wired.com/2015/08/pro-government-twitter-bots-try-hush-mexican-activists/.

Fino-Radin, Ben. In conversation with the author. Phone call, February 1, 2016.

F-Secure. "Brain: Searching for the First PC Virus in Pakistan." YouTube, March 9, 2011. https://www.youtube.com/watch?v=lnedOWfPKT0.

———. "From Brain to Stuxnet: 25 Years of PC Viruses." YouTube, February 2, 2011.
https://www.youtube.com/watch?v=d8Bpl-BUp0g.

Galloway, Alexander R., and Eugene Thacker. *The Exploit: A Theory of Networks*. Electronic
Mediations, v. 21. Minneapolis: University of Minnesota Press, 2007.

Garfinkel, Simson. "Re: Research on Malware," March 8, 2015.

Garza, George. "Top 10 Worst Computer Viruses." Catalogs.com. Accessed July 23, 2017.
http://www.catalogs.com/info/travel-vacations/top-10-worst-computer-viruses.html.

G DATA Software AG. "History of Malware." G DATA. Accessed March 22, 2016.
https://www.gdata-software.com/security-labs/information/history-of-malware.

Gordon, Sarah. "Inside the Mind of Dark Avenger." VX Heavens, January 1993.
https://download.adamas.ai/dlbase/Stuff/VX%20Heavens%20Library/static/vdat/ivdarkav.htm.

Graziano, Mariano, Corrado Leita, and Davide Balzarotti. "Towards Network Containment in
Malware Analysis Systems," 339. ACM Press, 2012.
https://doi.org/10.1145/2420950.2421000.

Greenberg, Andy. "This Artist's Images Integrate Code From Malware Like Stuxnet and Flame."
WIRED, November 27, 2014. http://www.wired.com/2014/11/malware-art/#slide-1.

Gruning, Jane. "Rethinking Viruses in the Archives." Poster presented at the Archival Education
and Research Institute, 2012.
https://www.ischool.utexas.edu/~janegru/images/Gruning_AERI2012.pdf.

"Hacker Lexicon: A Guide to Ransomware, the Scary Hack That's on the Rise." WIRED,
September 2015.
http://www.wired.com/2015/09/hacker-lexicon-guide-ransomware-scary-hack-thats-rise/#slide-2.

Hedstrom, Margaret L., Christopher A. Lee, Judith S. Olson, and Clifford A. Lampe. "'The Old
Version Flickers More': Digital Preservation from the User's Perspective." *The American
Archivist* 69 (Spring/Summer 2006): 159–87.

Honan, Mat. "Why Hackers Write Computer Viruses." Gizmodo, August 4, 2011.
http://gizmodo.com/5827405/why-hackers-write-computer-viruses.

Hypponen, Mikko. "Re: Malware Museum and Malware Preservation," April 7, 2016.

Hypponen, Mikko, and Peter Szor. "Spanska Threat Description." F-Secure, 1997.
https://www.f-secure.com/v-descs/spanska.shtml.

"Intervasion of the UK." Wikipedia. Accessed March 8, 2015.
http://en.wikipedia.org/wiki/Intervasion_of_the_UK.

John, Jeremy Leighton. "Digital Forensics and Preservation." Digital Preservation Coalition,
November 1, 2012.
http://www.dpconline.org/component/docman/doc_download/810-dpctw12-03pdf.

Jordan, Tim, and Paul A. Taylor. *Hacktivism and Cyberwars: Rebels with a Cause?* 1st ed.
London: Routledge, 2004.

J.P. Dyson. "So How DO You Preserve a Video Game?" presented at the Pressing Restart: Community Discussions on Video Game Preservation, NYU Game Center, September 28, 2013.

Jussi Parikka, and Tony D. Sampson, eds. *The Spam Book: On Viruses, Porn, and Other Anomalies from the Dark Side of Digital Culture*. Hampton Press Communication Series : Communication Alternatives. Cresskill, N.J: Hampton Press, 2009.

Kim, Julia. "Capturing a Shadow: Digital Forensics Applications with Born-Digital Legacy Material." *NDSR-NY* (blog), October 17, 2014. http://ndsr.nycdigital.org/capturing-a-shadow-digital-forensics-applications-with-born-digital-legacy-material/.

Kirschenbaum, Matthew. "Hello Worlds." *The Chronicle of Higher Education*, January 23, 2009. http://www.chronicle.com/article/Hello-Worlds/5476.

———. *Mechanisms: New Media and the Forensic Imagination*. Cambridge, Mass.: MIT Press, 2008.

Kirschenbaum, Matthew G., Richard Ovenden, Gabriela Redwine, and Rachel Donahue. *Digital Forensics and Born-Digital Content in Cultural Heritage Collections*. CLIR Publication, no. 149. Washington, D.C: Council on Library and Information Resources, 2010.

Knight, Gareth. "The Forensic Curator: Digital Forensics as a Solution to Addressing the Curatorial Challenges Posed by Personal Digital Archives." *International Journal of Digital Curation* 7, no. 2 (December 6, 2012): 40–63. https://doi.org/10.2218/ijdc.v7i2.228.

Korolov, Maria. "Significant Virtual Machine Vulnerability Has Been Hiding in Floppy Disk Code for 11 Years." CSO Online, May 13, 2015.

http://www.csoonline.com/article/2921589/application-security/significant-virtual-machine-vulnerability-has-been-hiding-in-floppy-disk-code-for-11-years.html.

Lamb, Steve. In conversation with the author, April 12, 2015.

Lowood, Henry. "Shall We Play a Game: Thoughts on the Computer Game Archive of the Future," October 2002. http://web.stanford.edu/~lowood/Texts/shall_game.pdf.

Ludovico, Alessandro. "Virus Charms and Self-Creating Codes." digitalcraft, n.d. http://www.digitalcraft.org/iloveyou/catalogue_alessandro_ludovico_virus_charms.htm.

Ludwig, Mark A. *The Little Black Book of Computer Viruses*. Tucson, Ariz: American Eagle Publications, 1991.

"Macro Virus." *Wikipedia, the Free Encyclopedia*, April 4, 2016. https://en.wikipedia.org/w/index.php?title=Macro_virus&oldid=713521000.

"Malware." Wikipedia. Accessed May 13, 2015. http://en.wikipedia.org/wiki/Malware.
Malware Example: CRASH.COM. MS-DOS, 2016. http://archive.org/details/malware_CRASH.COM.

Marquis-Boire, Morgan, Bill Marczak, Claudio Guarnieri, and John Scott-Railton. "You Only Click Twice: FinFisher's Global Proliferation." The Citizen Lab, March 13, 2013. https://citizenlab.org/2013/03/you-only-click-twice-finfishers-global-proliferation-2/.

McAllister, Neil. "Confession: I Was a Teenage Computer Virus Writer." The Register, September 14, 2015. http://www.theregister.co.uk/2015/09/14/i_was_a_teenage_virus_author/.

McLeod, Julie, and Catherine Hare, eds. *Managing Electronic Records*. London: Facet, 2005.

Mennerich, Don. In conversation with the author, December 3, 2015.

"Michelangelo (computer Virus)." Wikipedia. Accessed May 1, 2016.
    https://en.wikipedia.org/wiki/Michelangelo_(computer_virus).

Misra, Amit. "Antivirus Software Industry Growing, Despite Reports of Decline." Dazeinfo,
    August 25, 2015.
    http://dazeinfo.com/2015/08/25/antivirus-software-industry-growing-despite-reports-of-decli
    ne/.

"Morris Worm." Wikipedia. Accessed May 13, 2015. http://en.wikipedia.org/wiki/Morris_worm.

Nagy, Attila. "14 Infamous Computer Virus Snippets That Trace A History Of Havoc." Gizmodo
    Australia, July 6, 2013.
    http://www.gizmodo.com.au/2013/07/14-infamous-computer-virus-snippets-that-trace-a-hist
    ory-of-havoc/.

Nori, Franziska. "A Decade of Web Design." digitalcraft, January 2005.
    http://www.digitalcraft.org/index.php?artikel_id=550.

———. "I Love You." digitalcraft, 2002. http://www.digitalcraft.org/?artikel_id=284.
"OpenVMS." Wikipedia. Accessed March 8, 2015.
    http://en.wikipedia.org/wiki/OpenVMS#Major_release_timeline.

Parikka, Jussi. "Computer Viruses Deserve a Museum: They're an Art Form of Their Own." The
    Conversation, February 19, 2016.

https://theconversation.com/computer-viruses-deserve-a-museum-theyre-an-art-form-of-their-own-54762.

———. *Digital Contagions: A Media Archaeology of Computer Viruses*. Digital Formations, v. 44. New York: Peter Lang, 2007.

———. "RE: Query," February 19, 2016.

Pennock, Maureen. "Web Archiving." Digital Preservation Coalition, March 2013. http://dx.doi.org/10.7207/twr13-01.

Perlroth, Nicole. "Researchers Track Tricky Payment Theft Scheme." New York Times, November 24, 2015. http://bits.blogs.nytimes.com/2015/11/24/researchers-track-tricky-payment-theft-scheme/?_r=0.

Peterson, Christie. "Re: Malware Preservation," March 14, 2016.

Phillips, Megan, Jefferson Bailey, Andrea Goethals, and Trevor Owens. "The NDSA Levels of Digital Preservation: An Explanation and Uses." Library of Congress, n.d. http://www.digitalpreservation.gov/documents/NDSA_Levels_Archiving_2013.pdf.

PREMIS Editorial Committee. "Data Dictionary for Preservation Metadata: PREMIS Version 3.0." Library of Congress, June 2015. http://www.loc.gov/standards/premis/v3/premis-3-0-final.pdf.

"Rebel! Virus (1989) - Tommaso Tozzi." Accessed March 17, 2016. http://www.tommasotozzi.it/index.php?title=Rebel!_Virus_(1989).

"Recommended Formats Statement – Software and Electronic Gaming and Learning." Library of
        Congress. Accessed February 14, 2016.
        http://www.loc.gov/preservation/resources/rfs/softgame.html.

"Report: Average of 82,000 New Malware Threats per Day in 2013." PCWorld. Accessed April
        29, 2016.
        http://www.pcworld.com/article/2109210/report-average-of-82-000-new-malware-threats-pe
        r-day-in-2013.html.

"Rise of the Hackers." NOVA. Accessed April 20, 2016.
        http://www.pbs.org/wgbh/nova/tech/rise-of-the-hackers.html.

Rosenthal, David S. H. "Emulation & Virtualization as Preservation Strategies." Andrew W.
        Mellon Foundation, 2015.
        https://mellon.org/media/filer_public/0c/3e/0c3eee7d-4166-4ba6-a767-6b42e6a1c2a7/rosent
        hal-emulation-2015.pdf.

———. In conversation with the author. Phone call, April 19, 2016.

———. "The Malware Museum." *DSHR's Blog* (blog), February 9, 2016.
        http://blog.dshr.org/2016/02/the-malware-museum.html.

Rovelli, Paolo. "Don't Believe These Four Myths about Linux Security." *Sophos News* (blog),
        March 26, 2015.
        http://news.sophos.com/en-us/2015/03/26/dont-believe-these-four-myths-about-linux-securit
        y/.

Siluk, Shirley. "Internet Archive Displays Viruses of Past in Malware Museum." Sci-Tech
    Today, February 8, 2016.
    http://www.sci-tech-today.com/story.xhtml?story_id=0200028EPTUC.

Simpson, Justin. "Re: Research on Archivematica," March 28, 2016.

Sparrow, Jeff, and Jill Sparrow. *The Enemy within*. Radical Melbourne 2. Carlton North, Vic:
    Vulgar Press, 2004.

"SQL Slammer." Wikipedia. Accessed May 1, 2015.
    https://en.wikipedia.org/wiki/SQL_Slammer.

Stalbaum, Brett. "The Zapatista Tactical FloodNet." Electronic Civil Disobedience. Accessed
    May 1, 2016. http://www.thing.net/~rdom/ecd/ZapTact.html.

Stockley, Mark. "The Web Attacks That Refuse to Die." Naked Security, June 15, 2016.
    https://nakedsecurity.sophos.com/2016/06/15/the-web-attacks-that-refuse-to-die/.

"Stoned (computer Virus)." Wikipedia. Accessed March 22, 2016.
    https://en.wikipedia.org/wiki/Stoned_(computer_virus).

Stoner, Ed. "RE: Malware Preservation Research," March 29, 2016.

"Stuxnet." Wikipedia. Accessed December 6, 2015. https://en.wikipedia.org/wiki/Stuxnet.

Sullivan, Bob. "FBI Software Cracks Encryption Wall." NBC News, November 20, 2001.
    http://www.nbcnews.com/id/3341694/ns/technology_and_science-security/t/fbi-software-cra
    cks-encryption-wall/#.VyWF9KODGko.

Sumra, Husain. "'Hacking Team' Data Breach Confirms Firm's Ability to Infiltrate Jailbroken iPhones." Mac Rumors, July 6, 2015. http://www.macrumors.com/2015/07/06/hacking-team-jailbroken-iphone/.

swissnex San Francisco. "Project Cyber Virus: Digital Security Then and Now." swissnex San Francisco. Accessed January 5, 2016. http://www.swissnexsanfrancisco.org/event/projectcybervirusexhibit/.

———. "Project Cyber Virus: Opening Reception." Swissnex San Francisco. Accessed January 5, 2016. http://www.swissnexsanfrancisco.org/event/cybervirusopening/.

"The History and the Evolution of Computer Viruses: 2003-2008." *Privacy PC* (blog), March 25, 2012. http://privacy-pc.com/articles/the-history-and-the-evolution-of-computer-viruses-2003-2008.html.

Thomas A. Longstaff, E. Eugene Schultz. "Beyond Preliminary Analysis of the WANK and OILZ Worms: A Case Study of Malicious Code." *Computers & Security* 12, no. 1 (1993): 61–77. https://doi.org/10.1016/0167-4048(93)90013-U.

Thomas, Douglas. *Hacker Culture*. Minneapolis: University of Minnesota Press, 2002.

TIME. "Game Changers: Jim Lindner, Archive Automator." YouTube, March 23, 2012. https://www.youtube.com/watch?v=b8QvfimOfko.

"Top Ten Most Destructive Computer Viruses of All Time." Crunkish. Accessed May 1, 2016. http://crunkish.com/top-ten-worst-computer-viruses/.

University of Toronto, Munk School of Global Affairs. "The Citizen Lab." Accessed April 25, 2015. https://citizenlab.org/.

"VENOM Vulnerability." CrowdStrike. Accessed April 28, 2016.
http://venom.crowdstrike.com/.

Voon, Claire. "A Museum for the Blocky Graphics of Early Computer Viruses." Hyperallergic,
February 18, 2016.
https://hyperallergic.com/274139/a-museum-for-the-blocky-graphics-of-early-computer-viruses/.

"WANK (computer Worm)." Wikipedia. Accessed February 22, 2015.
http://en.wikipedia.org/wiki/WANK_%28computer_worm%29.

Wark, McKenzie. *A Hacker Manifesto*. Cambridge, MA: Harvard University Press, 2004.

White, Daniel. "Re: Malware Preservation Research," April 4, 2016.

Woods, Kam. "RE: Preserving Malware and EO1," April 25, 2016.

Woods, Kam, Christopher Lee, and Simson Garfinkel. "Extending Digital Repository
Architectures to Support Disk Image Preservation and Access," 2011.
http://www.ils.unc.edu/callee/p57-woods.pdf.

Zeltser, Lenny. "How Security Companies Assign Names to Malware Specimens." *Zeltser
Security Corp* (blog), October 26, 2011. https://zeltser.com/malware-naming-approaches/.